

Vergleich von Methoden zur Lösung der zeitabhängigen Schrödingergleichung

Diplomarbeit

von
Marc Hohl
aus
Regensburg

durchgeführt am Institut für Physikalische
und Theoretische Chemie unter Anleitung
von PD Dr. H. H. H. Homeier

November 1999

Inhaltsverzeichnis

1	Einführung	1
1.1	Zielsetzung	1
1.2	Thematik	1
1.3	Matrixdarstellungen des Hamiltonoperators	2
1.3.1	Basissatzverfahren	2
1.3.2	Gitterverfahren	4
1.4	Notationen und Konventionen	7
2	Zur Berechnung der Matrixexponentialfunktion	9
2.1	Polynomiale Approximationen	9
2.1.1	Taylor-Entwicklung	9
2.1.2	Cayley-Hamilton-Verfahren	10
2.1.3	Tschebyscheff-Approximation	11
2.2	Rationale Approximationen	14
2.2.1	Motivation	14
2.2.2	Padé-Approximationen	15
2.3	Matrixextrapolationsverfahren	16
2.3.1	ϵ -Algorithmus und verwandte Verfahren	17
2.3.2	Levin-artige Verfahren	18
2.4	Vektorextrapolationsverfahren	19
2.4.1	ϵ -Algorithmen	20
2.4.2	Projektionsmethoden	21
2.4.3	Vektor- \mathcal{J} -Transformation	21
2.5	Weitere Verfahren	22
2.5.1	Diagonalisierung	22
2.5.2	Programme zur Lösung von Differentialgleichungen	23
2.5.3	Cauchysche Integralformel	23
3	Verfahren zur Propagation von Wellenpaketen	25
3.1	Polynomiale Approximationen	25
3.1.1	Taylor-Reihe	25
3.1.2	Cayley-Hamilton-Verfahren	26
3.1.3	Tschebyscheff-Methode	26
3.2	Extrapolationsverfahren mit Matrixinversion	27
3.3	Extrapolationsverfahren ohne Matrixinversion	27
3.3.1	ϵ -Algorithmen	27
3.3.2	Projektionsmethoden	28
3.3.3	Vektor- \mathcal{J} -Transformation	28
3.4	Split-Operator-Verfahren	29

4	Numerische Tests	32
4.1	Berechnung der Matrixexponentialfunktion	32
4.1.1	Allgemeines	32
4.1.2	Tschebyscheff-Approximation	33
4.1.3	Taylorapproximation	37
4.1.4	Vergleich von Tschebyscheff- und Taylor-Approximation	41
4.1.5	Beschleunigungsverfahren	43
4.2	Vektorpropagation	51
4.2.1	Vorbemerkungen	51
4.2.2	Tschebyscheff-Approximation	52
4.2.3	Beschleunigungsverfahren	52
4.2.4	Vergleich von Tschebyscheff- und Beschleunigungsverfahren	54
5	Anwendungen auf Modellsysteme	56
5.1	Die Zeitentwicklung von I_2	56
5.1.1	Motivation und physikalische Grundlagen	56
5.1.2	Diskretisierung durch DVR	58
5.1.3	Propagationsverfahren	59
5.2	Propagation durch SPO-Verfahren	66
5.3	Das van der Waals-Molekül HeI_2	68
5.3.1	Physikalische Grundlagen	68
5.3.2	Diskretisierung durch DVR	70
5.3.3	Die Wellenfunktion ψ_0	72
5.3.4	Wahl des Koordinatenbereiches	73
5.4	Propagation und Ergebnisse	74
6	Zusammenfassung	80
A	Algorithmen	82
A.1	Vektorieller ϵ -Algorithmus	82
A.2	Topologische ϵ -Algorithmen	83
A.3	\mathcal{J} -Algorithmen	85
B	Das DVR-Verfahren	87
B.1	Formaler Zugang	87
B.2	Ein universelles DVR-Verfahren	89
C	Optische Potentiale	92

1 Einführung

1.1 Zielsetzung

Ziel der Arbeit ist es, Extrapolationsverfahren zur Lösung der zeitabhängigen Schrödinger-Gleichung mit aus der Literatur bekannten Methoden zu vergleichen. Die anhand von numerischen Tests entwickelten Programme sollen auf Modellsysteme angewandt werden.

1.2 Thematik

Ausgangspunkt ist die zeitabhängige Schrödinger-Gleichung¹

$$i \frac{\partial}{\partial t} \psi(t) = \hat{H} \psi(t) \quad (1)$$

mit dem Anfangswert $\psi_0 = \psi(0)$. Durch Einführung einer Basis (siehe 1.3.1) oder durch Diskretisierung der räumlichen Freiheitsgrade (siehe 1.3.2) läßt sich \mathbf{H} als Matrix und $\psi(t)$ als Vektor auffassen.

Für zeitabhängige Hamiltonoperatoren schreibt man

$$\psi(t) = \hat{U}(t) \psi_0, \quad (2)$$

wobei $\hat{U}(t)$ den Zeitentwicklungsoperator bezeichnet.

Ist der Hamiltonoperator \hat{H} zeitunabhängig, so läßt sich die Lösung $\psi(t)$ der Schrödinger-Gleichung durch Anwendung des Zeitentwicklungsoperators $\exp(-i \mathbf{H} t)$ auf den Anfangswert ψ_0 berechnen:

$$\psi(t) = e^{-i \mathbf{H} t} \psi_0. \quad (3)$$

Damit ist die Berechnung der Zeitentwicklung der Wellenfunktion formal gelöst. In der Praxis ist die numerische Lösung von (1) ein wichtiges Problem. Laut [1, S. 103] sind „Algorithmen zu deren Lösung ... heute noch Gegenstand aktueller Forschungsarbeiten“.

In der vorliegenden Arbeit werden zwei grundsätzliche Methoden zur Lösung der Schrödinger-Gleichung vorgestellt:

Zum einen ist dies die Approximation des Zeitentwicklungsoperators über die Matrix-exponentialfunktion und dessen Anwendung auf die Wellenfunktion, um deren zeitliche Propagation zu beschreiben. Dieses Verfahren hat den Vorteil, daß der Zeitentwicklungsoperator nur einmal berechnet werden muß und dann auf verschiedene Wellenfunktionen bzw. Wellenpakete angewendet werden kann. Ersetzt man den Zeitpunkt t in (3) durch ein Zeitintervall Δt , so erhält der Zeitentwicklungsoperator die Form $\exp(-i \mathbf{H} \Delta t)$. Durch n -maliges Anwenden dieses Operators auf ψ_0 erhält man die Folge

$$\psi(0), \psi(\Delta t), \psi(2 \cdot \Delta t), \dots \psi(n \cdot \Delta t), \quad (4)$$

¹Im folgenden ist der Einfachheit halber immer $\hbar = 1$ gesetzt, siehe auch Abschnitt 1.4.

die den zeitlichen Verlauf der Wellenfunktion veranschaulicht. Kap. 2 beschäftigt sich mit derartigen Verfahren.

Diese Methoden eignen sich aber nicht nur zur Bestimmung des Zeitentwicklungsoperators. Weitere Einsatzmöglichkeiten liegen z. B. im Übergang von der Schrödinger- zur Wechselwirkungs-Darstellung in der Quantenmechanik, der durch einen Operator der Form $\exp(i\hat{H}_0 t)$ beschrieben wird, wobei $\hat{H} = \hat{H}_0 + \hat{V}$ gilt (siehe auch Gl. (149) in Abschnitt 3.4).

Auch der thermodynamische Dichteoperator ist ein Operatorexponential:

$$\hat{\rho} = \frac{e^{-\beta\hat{H}}}{\text{tr}(e^{-\beta\hat{H}})}. \quad (5)$$

Bei geeigneter Basiswahl läßt sich ρ als Matrixexponentialfunktion darstellen. Allgemeine Differentialgleichungen der Form

$$\frac{d}{dt}\vec{v}(t) = \mathbf{A}(t)\vec{v}(t), \quad \vec{v}(0) = \vec{v}_0 \quad (6)$$

mit einer (u. U. zeitabhängigen) Matrix \mathbf{A} führen auf Beziehungen der Form

$$\vec{v}(t + \Delta t) = e^{\mathbf{A}(t)\cdot\Delta t}\vec{v}(t). \quad (7)$$

Ist man dagegen nur am Endzustand $\psi(t)$ interessiert, bieten sich als zweite Möglichkeit Vektorverfahren an, die eine Folge der Vektoren

$$\psi_0 = \psi(0), \psi_1, \dots, \psi_n, \dots, \psi_\infty = \psi(t) \quad (8)$$

extrapolieren. Diese Vorgehensweise wird in Kap. 3 genauer ausgeführt. Der Vorteil hierbei liegt in einem geringeren Rechenaufwand, da der Endzustand nicht auf dem Umweg über ein hinreichend genau approximiertes Matrixexponential berechnet, sondern direkt aus einer Folge von Vektoren bestimmt wird.

1.3 Matrixdarstellungen des Hamiltonoperators

Im folgenden werden konkrete Verfahren vorgestellt, die den Hamiltonoperator in eine Matrix überführen, um so die Grundlage für numerische Berechnungen zu liefern.

1.3.1 Basissatzverfahren

Gegeben sei eine Menge von Funktionen

$$\phi_j(x), \quad j = 1, \dots, \infty, \quad (9)$$

die orthonormal sind, d. h. die

$$\langle \phi_j(x), \phi_k(x) \rangle = \delta_{jk} \quad (10)$$

erfüllen, und die darüberhinaus vollständig ist, was sich in der Form

$$\sum_{j=0}^{\infty} |\phi_j(x)\rangle \langle \phi_j(x)| = \hat{1} \quad (11)$$

darstellen läßt. Sind (10) und (11) erfüllt, bildet (9) eine *Basis* des Hilbertraums. Im folgenden wird statt $\phi_j(x)$ der Einfachheit halber immer ϕ_j geschrieben.

Nun läßt sich jeder Operator \hat{A} in dieser Basis darstellen gemäß

$$\mathbf{A} = (A_{jk}) = \langle \phi_j, \hat{A}\phi_k \rangle, \quad j, k = 1, \dots, \infty, \quad (12)$$

d. h. aus dem Operator \hat{A} wird eine (unendliche) Matrix $\mathbf{A} = (A_{jk})$.

Ebenso läßt sich die Wellenfunktion $\psi(x, t)$ nach dieser Basis entwickeln:

$$\psi(x, t) = \sum_{j=0}^{\infty} \langle \phi_j, \psi(x, t) \rangle \phi_j = \sum_{j=0}^{\infty} c_j(t) \phi_j(x); \quad (13)$$

hierbei bilden die Koeffizienten

$$c_j(t) = \langle \phi_j, \psi(x, t) \rangle \quad (14)$$

einen i. a. zeitabhängigen, unendlichdimensionalen Vektor.

Da mit unendlichdimensionalen Matrizen und Vektoren keine numerischen Berechnungen durchgeführt werden können, reduziert man die unendliche Basis auf eine Menge von D Funktionen,

$$\{\phi_j; j = 1, \dots, D\}, \quad (15)$$

die so gewählt ist, daß die Beiträge A_{jk} und c_j für $j, k > D$ im Rahmen der Rechengenauigkeit vernachlässigbar sind. Es müssen also endlichdimensionale *Basisätze* gefunden werden, die bei möglichst kleiner Dimension D die wesentlichen Aspekte einer gegebenen Problemstellung erfassen.

Schneidet man die Entwicklungen (12) und (13) nach $j, k = D$ ab, so erhält man aus den Operatoren ($D \times D$)-Matrizen

$$\mathbf{A} = (A_{jk}) = \langle \phi_j, \hat{A}\phi_k \rangle, \quad j, k = 1, \dots, D \quad (16)$$

und aus der Wellenfunktion D -dimensionale Vektoren:

$$\psi(x, t) \approx \sum_{j=0}^D c_j(t) \phi_j(x). \quad (17)$$

1.3.2 Gitterverfahren

Grundsätzlich läßt sich der Raum durch die Einführung eines Gitters diskretisieren. Der Ausgangszustand ψ_0 wird dann durch seine Werte an den Orten der Gitterpunkte beschrieben und somit zu einem Vektor. Dabei muß Gittergröße und -auflösung nach folgenden Gesichtspunkten gewählt werden:

- Da das Gitter nur einen endlichen Raumbereich überdecken kann, ist die Wahl der Grenzen entscheidend. Erreichen Teile der Wellenfunktion im beobachteten Zeitraum die Grenzen, so werden sie dort entweder reflektiert oder (bei der Verwendung von Fourier-Transformationen, s. u.) auf die gegenüberliegende Seite des Gitters projiziert. Beides sind unerwünschte Effekte, die die Genauigkeit und den Aussagegehalt der Berechnungen empfindlich stören können.
- Die Auflösung des Gitters bestimmt direkt die unterste Schranke der beobachtbaren Veränderungen. Ein Maß für die Ortsauflösung ist die de Broglie-Wellenlänge der Wellenfunktion.
- Die Größe und die Auflösung des Gitters bestimmen auch die Grenzen der darstellbaren Energie und sollten deshalb an das Spektrum des Hamiltonoperators angepaßt sein.
- Die Anzahl der Gitterpunkte D bestimmt den Rechenaufwand des Problems. Man beachte, daß die Hamiltonoperator durch die Diskretisierung des Raumes zu einer $(D \times D)$ -Matrix wird. Bei Problemstellungen mit mehreren Freiheitsgraden und hoher Ortsauflösung stößt man hier sehr schnell an die Grenzen von Rechnerleistung und Speicherbedarf (vgl. dazu die Anmerkungen in Abschnitt 5.1.3).

Ist ψ_0 diskretisiert, so muß $\hat{H}\psi_0$ berechnet werden. Üblicherweise kann der Hamiltonoperator in einen Operator \hat{T} der kinetischen Energie und einen Operator \hat{V} der potentiellen Energie aufgespalten werden. Während die Anwendung von \hat{V} auf ψ_0 im wesentlichen einer Multiplikation mit V an jedem Gitterpunkt entspricht (\hat{V} ist diagonal in der Ortsdarstellung), ist die Berechnung von $\hat{T}\psi_0$ problematischer, da \hat{T} in dieser Darstellung nichtlokal ist.

Eine Möglichkeit, die zweiten Ableitungen von ψ_0 zu berechnen, ist die Verwendung von *finiten Differenzen* (FD). In [2, 3, 4] wird ein 3-Punkt FD-Verfahren verwendet, das die zweite Ableitung wie folgt approximiert (hier im eindimensionalen Fall):

$$\frac{d^2\psi_0(x_\alpha)}{dx^2} = \frac{\psi_0(x_{\alpha+1}) - 2\psi_0(x_\alpha) + \psi_0(x_{\alpha-1}))}{(\Delta x)^2} \quad (18)$$

Hier wurde angenommen, daß die Punkte x_α , $\alpha = 1, \dots, D$ im Abstand Δx angeordnet sind. Ein Beispiel für den Einsatz einer 5-Punkt FD-Formel findet sich in [5]. In [6, Anh. B] ist die allgemeine Formel für ein N -Punkt FD-Verfahren und der Grenzübergang $N \rightarrow \infty$ hergeleitet. Die Verwendung von finiten Differenzen stellt (für $N < \infty$)

eine semilokale Näherung eines nichtlokalen Operators dar und ist daher u. U. stark fehlerbehaftet (vgl. dazu auch [7]).

Ein weiteres Verfahren zur Diskretisierung von Operatoren und Wellenfunktionen, die *discrete variable representation* (DVR) besteht darin, im Raum D Punkte $\{x_\alpha, \alpha = 1, \dots, D\}$ festzulegen. Jedem dieser Punkte x_α ist ein Gewichtungsfaktor ω_α zugeteilt. Außerdem wird wie in Kap. 1.3.1 eine D -dimensionale Basis

$$\{\phi_j; j = 1, \dots, D\} \quad (19)$$

verwendet.

Die Wellenfunktion läßt sich wieder in der Form

$$\psi(x, t) = \sum_{j=1}^D d_j(t) \phi_j(x) \quad (20)$$

darstellen (vgl. (17)), wobei hier zur Unterscheidung statt der Koeffizienten c_j die diskreten Koeffizienten d_j verwendet wurden. Es gilt nämlich

$$d_j(t) = \sum_{\alpha=1}^D \omega_\alpha \phi_j^*(x_\alpha) \psi(x_\alpha, t) \quad (21)$$

im Gegensatz zu (14), wenn die ϕ_j auch in der diskreten Darstellung noch orthonormal sind in dem Sinne, daß

$$\sum_{\alpha=1}^D \omega_\alpha \phi_j^*(x_\alpha) \phi_k(x_\alpha) = \delta_{jk} \quad (22)$$

gilt. Formal approximiert man also die Integration durch eine gewichtete Summation über die D Gitterpunkte:

$$\int dx \phi_j^*(x) \dots \longrightarrow \sum_{\alpha=1}^D \omega_\alpha \phi_j^*(x_\alpha) \dots \quad (23)$$

Es läßt sich zeigen, daß die DVR-Darstellung durch eine orthogonale Transformation aus der Basissatzdarstellung nach 1.3.1 hervorgeht [8, 9, 10]. Der ersichtliche Vorteil der DVR-Methode ist der, daß statt der Berechnung von Integralen über den Ortsraum nur Summationen über die Gitterpunkte durchgeführt werden müssen.

Setzt man (21) in (20) ein, so erhält man

$$\psi(x, t) = \sum_{j=1}^D \sum_{\alpha=1}^D \omega_\alpha \phi_j^*(x_\alpha) \psi(x_\alpha, t) \phi_j(x) \quad (24)$$

$$= \sum_{\alpha=1}^D \psi_\alpha \chi_\alpha(x); \quad (25)$$

hierbei wurden die Funktionen

$$\chi_\alpha(x) = \sqrt{\omega_\alpha} \sum_{j=1}^D \phi_j^*(x_\alpha) \phi_j(x), \quad j = 1, \dots, D \quad (26)$$

eingeführt, die eine orthonormale Basis in der diskreten Darstellung bilden, sowie die Koeffizienten

$$\psi_\alpha = \sqrt{\omega_\alpha} \psi(x_\alpha, t) \quad \alpha = 1, \dots, D. \quad (27)$$

Die n -ten Ableitungen der Wellenfunktion nach den Ortskoordinaten (z. B. $n = 2$ für die Anwendung des Operators der kinetischen Energie) berechnet man gemäß

$$\frac{\partial^n \psi(x, t)}{\partial x^n} = \sum_{\alpha=1}^D \psi_\alpha \frac{\partial^n \chi_\alpha(x)}{\partial x^n}. \quad (28)$$

Dabei können die Ableitungen

$$\frac{\partial^n \chi_\alpha(x)}{\partial x^n} = \sqrt{\omega_\alpha} \sum_{j=1}^D \phi_j^*(x_\alpha) \frac{\partial^n \phi_j(x)}{\partial x^n} \quad (29)$$

für alle x_α ($\alpha = 1, \dots, D$) einmal berechnet und dann abgespeichert werden.

Ein Sonderfall der DVR ist die Verwendung schneller Fourier-Transformationen (FFT) [11]. Man erhält dieses Verfahren aus dem oben beschriebenen, in dem man die Gitterpunkte x_α in einheitlichen Abstand voneinander setzt, also $x_\alpha = \alpha \Delta x$ ($\alpha = 0, \dots, D - 1$) mit der Gitterkonstanten Δx (hier im eindimensionalen Fall) wählt. Darüberhinaus werden als Basisfunktionen ebene Wellen der Form $\phi_\alpha(x) = \exp(ik_\alpha x)$ ($\alpha = 0, \dots, D - 1$) eingesetzt. Es zeigt sich, daß sich hier die Anwendung des Operators der kinetischen Energie auf eine Fouriertransformation der Wellenfunktion in den Impulsraum, einer Multiplikation mit $-k_\alpha^2$ und einer anschließenden Rücktransformation in den Ortsraum reduzieren läßt (vgl. auch Kap. 3.4).

Im Vergleich von DVR mit der FFT-Methode zeigt sich, daß bei ersterer der Rechenaufwand proportional zu D^2 steigt, bei der Fourier-Methode dagegen nur mit $D \log_2 D$. Der Vorteil des allgemeinen DVR-Verfahrens liegt darin, daß die Wahl der Gitterpunkte und der verwendeten Basis speziell an das betrachtete Problem angepaßt werden kann (z. B. können die Gitterpunkte im Bereich starker Wechselwirkungen dichter liegen als in Randbereichen) und kein Wechsel zwischen Orts- und Impulsdarstellung notwendig ist.

In Anhang B.1 wird ein formaler Zugang zum DVR-Verfahren beschrieben, bei dem die Zahl der Basisfunktionen verschieden von der Zahl der Gitterpunkte sein kann. Anhang B.2 zeigt ein konkretes DVR-Verfahren, das im praktischen Teil dieser Arbeit (Kap. 5) eingesetzt wurde und in seiner Konzeption universell einsetzbar ist.

1.4 Notationen und Konventionen

Operatoren werden mit einem $\hat{}$ dargestellt, z. B. der Hamiltonoperator

$$\hat{H}, \quad (30)$$

die zugehörigen *Matrizen* werden ohne Akzent dargestellt, also

$$H. \quad (31)$$

Das *Spektrum* eines Operators \hat{X} , also die Menge seiner Eigenwerte², wird durch

$$\sigma(\hat{X}) \quad (33)$$

symbolisiert.

Um im folgenden die Formeln übersichtlicher zu halten, werden atomare Einheiten verwendet, d. h. es gilt:

$$\hbar = 1, \quad e = 1, \quad m_e = 1. \quad (34)$$

Das *Kroneckersymbol* δ_{jk} hat folgende Bedeutung:

$$\delta_{jk} = \begin{cases} 1 & \text{für } j = k \\ 0 & \text{für } j \neq k. \end{cases} \quad (35)$$

Der *Differenzenoperator* Δ wirkt, wenn nicht gesondert indiziert, auf den Index n gemäß

$$\Delta f(n) = f(n+1) - f(n). \quad (36)$$

Differenzen höherer Ordnung sind wie folgt definiert:

$$\Delta^{k+1} f(n) = \Delta^k f(n+1) - \Delta^k f(n). \quad (37)$$

Für $k = 1$ wird mit $\Delta^1 = \Delta$ daraus

$$\Delta^2 f(n) = \Delta f(n+1) - \Delta f(n) = f(n+2) - 2f(n+1) + f(n). \quad (38)$$

Die *Gammafunktion* $\Gamma(x)$ ist definiert durch [12]:

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt \quad (x > 0). \quad (39)$$

Die *Besselfunktionen* $J_n(x)$ *erster Art* der Ordnung n sind durch

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k \left(\frac{x}{2}\right)^{n+2k}}{k! \Gamma(n+k+1)} \quad (40)$$

²genauer: der *Abschluß* der Menge seiner Eigenwerte, also

$$\sigma(\hat{X}) = \overline{\{\lambda : \lambda \text{ ist Eigenwert von } \hat{X}\}} \quad (32)$$

definiert [12, S. 389, Gl. (9.52a)].

Die *Norm* einer Matrix ist, wenn nicht anders vermerkt, die 2-Norm, definiert durch

$$\|x\| = \sqrt{\sum_{j=1}^D |x_j|^2}, \quad \|A\| = \max_{\|x\|=1} \|Ax\|. \quad (41)$$

In den Computerprogrammen wird dagegen stets die *1-Norm* verwendet, die durch

$$\|x\|_1 = \max_{1 \leq j \leq D} |x_j|, \quad \|A\|_1 = \max_{1 \leq j \leq D} \sum_{i=0}^D |a_{ij}| \quad (42)$$

gegeben ist.

Das *Skalarprodukt* zweier Funktionen $\phi(x)$ und $\psi(x)$ ist definiert durch

$$\langle \phi(x, t), \psi(x, t) \rangle := \int_0^\infty \phi^*(x, t) \psi(x, t) dx. \quad (43)$$

Dabei steht x für die Gesamtheit der räumlichen Koordinaten, von denen ϕ und ψ abhängen.

2 Methoden zur Berechnung der Matrixexponentialfunktion

2.1 Polynomiale Approximationen

2.1.1 Taylor-Entwicklung

Die Exponentialfunktion einer beliebigen $(D \times D)$ -Matrix \mathbf{A} ist definiert durch die Taylorreihe der Exponentialfunktion mit Matrixargument:

$$e^{\mathbf{A}} := \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!}. \quad (44)$$

Dieser Ausdruck kann durch das Taylorpolynom $p_N(\mathbf{A})$ vom Grad N approximiert werden:

$$e^{\mathbf{A}} \approx p_N(\mathbf{A}) = \sum_{n=0}^N \frac{\mathbf{A}^n}{n!}. \quad (45)$$

Der Konvergenzradius der Taylorreihe ist unendlich, somit ist zumindest theoretisch die Exponentialfunktion beliebig genau berechenbar, wenn nur genügend viele Summanden aufaddiert werden. In der Praxis treten jedoch mehrere Probleme auf. In [13] wird darauf hingewiesen, daß die Taylorreihe schon im skalaren Fall sehr langsam konvergiert. Oft treten dabei sehr große Summanden mit alternierenden Vorzeichen auf, sodaß Rundungsfehler, bedingt durch die endliche Stellenzahl der Computer, sehr bedeutsam werden und unter Umständen zu völlig falschen Ergebnissen führen.

Das Konvergenzverhalten einer Matrix kann durch ihre Norm abgeschätzt werden. In [13] wird als Fehlerabschätzung die Ungleichung

$$\|p_N(\mathbf{A}) - e^{\mathbf{A}}\| \leq \left(\frac{\|\mathbf{A}\|^{N+1}}{(N+1)!} \right) \left(\frac{1}{1 - \|\mathbf{A}\|/(N+2)} \right) \leq \delta \quad (46)$$

angegeben. Will man also eine Fehlertoleranz von δ unterschreiten, so muß man N so wählen, daß (46) erfüllt ist. Man sieht, daß N umso größer sein muß, je größer die Norm $\|\mathbf{A}\|$ ist.

Man kann nun das Konvergenzverhalten verbessern und die Rundungsfehler minimieren, wenn man die Identität

$$e^{\mathbf{A}} = \left(e^{\mathbf{A}/s} \right)^s \quad (47)$$

verwendet [13, 14, 15]. Setzt man den Skalierungsfaktor $s = 2^q$, so erhält man die Exponentialmatrix durch q -faches Quadrieren von $e^{\mathbf{A}/s}$, wofür effiziente Computeralgorithmen existieren.

Die Matrizen in unserem speziellen Fall sind ja von der Form $\mathbf{A} = -i\mathbf{H}\Delta t$, sodaß sich für kleine Zeitschritte Δt die Matrixexponentialfunktion durch

$$e^{-i\mathbf{H}\Delta t} = 1 - i\mathbf{H}\Delta t + O(\Delta t^2), \quad (48)$$

also durch die führenden Terme der Taylorreihe nähern läßt. Diese Näherung ist allerdings nicht unitär und erhält somit nicht die Norm der Wellenfunktion, die propagiert werden soll.

Eine Verbesserung dieser Näherung durch rationale Funktionen wird in 2.2 diskutiert.

2.1.2 Cayley-Hamilton-Verfahren

Dieses Verfahren beruht auf der Kenntnis des charakteristischen Polynoms $P_A(z)$ [13]. Dies ist definiert durch

$$P_A(z) = \det(z\mathbf{1} - \mathbf{A}) = z^n - \sum_{k=0}^{n-1} c_k z^k. \quad (49)$$

Nach dem Satz von Cayley-Hamilton ist $P_A(\mathbf{A}) = 0$ und damit

$$\mathbf{A}^n = c_0\mathbf{1} + c_1\mathbf{A} + \dots + c_{n-1}\mathbf{A}^{n-1}, \quad (50)$$

d. h. jede Potenz von \mathbf{A} läßt sich als Linearkombination der $\mathbf{A}^0 = \mathbf{1}, \mathbf{A}, \dots, \mathbf{A}^{n-1}$ darstellen:

$$\mathbf{A}^k = \sum_{j=0}^{n-1} \beta_{kj} \mathbf{A}^j, \quad k = 0, 1, \dots \quad (51)$$

Somit kann e^{At} zu einem Polynom in \mathbf{A} mit analytischen Koeffizienten in t umgeformt werden:

$$\begin{aligned} e^{At} &= \sum_{k=0}^{\infty} \frac{\mathbf{A}^k t^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \left[\sum_{j=0}^{n-1} \beta_{kj} \mathbf{A}^j \right] \\ &= \sum_{j=0}^{n-1} \left[\sum_{k=0}^{\infty} \beta_{kj} \frac{t^k}{k!} \right] \mathbf{A}^j \\ &= \sum_{j=0}^{n-1} \alpha_j(t) \mathbf{A}^j. \end{aligned} \quad (52)$$

Die Koeffizienten β_{kj} , die die analytischen Koeffizienten gemäß $\alpha_j(t) = \sum_k \beta_{kj} t^k / k!$ bilden, lassen sich wie folgt rekursiv berechnen:

$$\beta_{kj} = \begin{cases} \delta_{kj} & (k < n) \\ c_j & (k = n) \\ c_0 \beta_{k-1, n-1} & (k > n, j = 0) \\ c_j \beta_{k-1, n-1} + \beta_{k-1, j-1} & (k > n, j > 0). \end{cases} \quad (53)$$

Bei der Rekursion der Koeffizienten können Rundungsfehler das Ergebnis stark beeinflussen [13]. Im skalaren Fall (1×1 -Matrix) reduziert sich der analytische Koeffizient zu einer Taylorreihe, und die dort (2.1.1) besprochenen Fehlerquellen kommen auch hier zum Tragen.

Ein weiterer Nachteil ist die Tatsache, daß das charakteristische Polynom der Matrix bekannt sein muß. Die Berechnung von $P_A(z)$ (beispielsweise über die Eigenwerte von A) ist sehr rechenintensiv und wiederum anfällig gegenüber Rundungsfehlern.

In [13] sind weitere Verfahren beschrieben, denen eine andere Basis des von den Matrizen A^0, A^1, \dots, A^{n-1} aufgespannten Unterraums zugrundeliegt. Die Nachteile entsprechen den oben genannten.

2.1.3 Tschebyscheff-Approximation

Eine polynomiale Entwicklung, die weniger rechenintensiv ist und darüberhinaus eine höhere Rechengenauigkeit aufweist [16], ist die Approximation der Exponentialfunktion durch Tschebyscheff-Polynome.

Es zeigt sich, daß sich die Berechnung von e^{-iA} auf die Approximation der skalaren Funktion e^{-iz} zurückführen läßt [16], wobei $z \in \sigma(A)$ ist. Der Bereich, in dem die z liegen, läßt sich durch eine Kreisscheibe um einen Mittelwert \bar{z} mit dem Radius Δz darstellen (siehe auch Abbildung 1):

$$z = \bar{z} + x\Delta z, \quad \text{wobei } z \in \sigma(A) \quad \text{und} \quad |x| \leq 1. \quad (54)$$

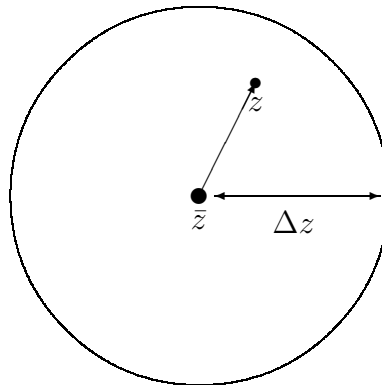


Abbildung 1: Zur Zerlegung von $z = \bar{z} + x\Delta z$.

In unserem speziellen Fall ist wegen $A = H\Delta t$ und der Hermitizität von H z reell; somit ist auch x reell:

$$x \in [-1, 1], \quad (55)$$

d. h. x läßt sich in der Form

$$x = \cos(\theta) \quad (56)$$

schreiben.

Aus Gleichung (54) folgt für e^{-iz} :

$$e^{-iz} = e^{-i\bar{z}} e^{-ix\Delta z}. \quad (57)$$

Mit [17, Gln. (9.1.44) und (9.1.45)] läßt sich $e^{\pm ix\Delta z}$ schreiben als

$$e^{\pm ix\Delta z} = J_0(\Delta z) + 2 \sum_{n=1}^{\infty} (\pm i)^n J_n(\Delta z) T_n(x). \quad (58)$$

Die Tschebyscheff-Polynome T_n sind dabei definiert als

$$T_n(\cos \theta) = \cos n\theta. \quad (59)$$

Gleichbedeutend damit ist die Definition [12, S. 782, Gl. (19.183a)]

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1, 1] \quad (60)$$

Somit folgt für e^{-iz} :

$$\begin{aligned} e^{-iz} &= e^{-i\bar{z}} e^{-ix\Delta z} \\ &= e^{-i\bar{z}} \left(J_0(\Delta z) + 2 \sum_{n=1}^{\infty} (-i)^n J_n(\Delta z) T_n(x) \right) \\ &= e^{-i\bar{z}} \sum_{n=0}^{\infty} (-i)^n C_n J_n(\Delta z) T_n(x) \end{aligned} \quad (61)$$

mit den Normierungsfaktoren

$$C_n = \begin{cases} 1 & \text{für } n = 0 \\ 2 & \text{sonst.} \end{cases} \quad (62)$$

und den Besselfunktionen J_n , siehe (40).

Für die Tschebyscheffpolynome gilt folgende Rekursionsformel [12, S. 782, Gl. (19.186)]:

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x \\ T_{n+1}(x) &= 2x \cdot T_n(x) - T_{n-1}(x) & (n \geq 1). \end{aligned} \quad (63)$$

Die Entwicklung (61) läßt sich nun approximieren, indem man nach einer endlichen Zahl N von Summanden die Summation abbricht:

$$e^{-iz} \approx \sum_{n=0}^N a_n T_n(x). \quad (64)$$

Die Definition der Entwicklungskoeffizienten folgt hierbei aus (61):

$$a_n = e^{-i\bar{z}} (-i)^n C_n J_n(\Delta z) \quad (65)$$

Hier wird auch ein großer Vorteil dieser Methode sichtbar: die Besselfunktion $J_n(\Delta z)$ geht für $n > \Delta z$ exponentiell gegen Null. Das bedeutet, daß nur wenige Terme oberhalb $n = \Delta z$ berücksichtigt werden müssen, um Konvergenz zu erreichen [7].

Für die konkrete Berechnung der Matrixexponentialfunktion $\exp(-i\mathbf{H}\Delta t)$ ist eine Abschätzung für den Betrag des größten und des kleinsten Eigenwerts von \mathbf{H} notwendig³. Erhält man die Matrixdarstellung von $\hat{\mathbf{H}}$ durch Einführung eines Raumgitters (siehe auch 1.3.2), so kann der Bereich der Eigenwerte von \mathbf{H} über die Extremalwerte der kinetischen und der potentiellen Energie näherungsweise bestimmt werden [16]: Der maximale Impuls p_{\max} , der auf einem Gitter dargestellt werden kann, ist mit dem Gitterabstand Δx verknüpft über

$$p_{\max} = \frac{\pi}{\Delta x}. \quad (66)$$

Daraus folgt ein Wert für die maximale kinetische Energie

$$T_{\max} = \frac{p_{\max}^2}{2m} = \frac{\pi^2}{2m\Delta x^2}. \quad (67)$$

Sind V_{\min} und V_{\max} die minimalen bzw. maximalen Werte der potentiellen Energie auf dem Gitter, so liegen die Eigenwerte von \mathbf{H} in dem Bereich

$$\sigma(\mathbf{H}) \subset \left[V_{\min}, V_{\max} + \frac{\pi^2}{2m\Delta x^2} \right]. \quad (68)$$

Aus den endlichen Abmessungen des Raumgitters resultiert ein minimaler Wert der kinetischen Energie, die sog. Nullpunktsenergie, die in obiger Abschätzung nicht berücksichtigt wurde. Somit ist das betrachtete Intervall etwas größer als der tatsächliche Spektralbereich von \mathbf{H} .

Folglich kann der Hamiltonoperator renormiert werden, sodaß

$$\sigma(\hat{\mathbf{H}}_{\text{norm}}) \subset [-1, 1] \quad (69)$$

gilt. Mit den Abkürzungen

$$\Delta E = \frac{1}{2} \left(\frac{\pi^2}{2m\Delta x^2} + V_{\max} - V_{\min} \right) \quad (70)$$

$$\bar{E} = \frac{1}{2} \left(\frac{\pi^2}{2m\Delta x^2} + V_{\max} + V_{\min} \right) \quad (71)$$

für den Bereich ΔE und den Mittelwert \bar{E} der Energieeigenwerte folgt

$$\hat{\mathbf{H}}_{\text{norm}} = \frac{\hat{\mathbf{H}} - \hat{\mathbf{1}}\bar{E}}{\Delta E} \quad (72)$$

für den renormierten Hamiltonoperator. Aus

$$\mathbf{H} = \hat{\mathbf{1}}\bar{E} + \mathbf{H}_{\text{norm}}\Delta E \quad (73)$$

³Bei anderen physikalischen Problemstellungen wäre z.B. auch eine Abschätzung über die *Gerschgorin-Kreise* möglich (siehe dazu auch [18, Kap. 2.3])

folgt

$$e^{-iH\Delta t} = e^{-i\bar{E}\Delta t} e^{-iH_{\text{norm}}\Delta E\Delta t} \quad (74)$$

in der Matrixschreibweise. Ein Vergleich mit (57) liefert die Substitutionen

$$z \rightarrow H\Delta t, \quad (75)$$

$$\bar{z} \rightarrow \bar{E}\Delta t, \quad (76)$$

$$\Delta z \rightarrow \Delta E\Delta t, \quad (77)$$

$$x \rightarrow H_{\text{norm}}. \quad (78)$$

Eingesetzt in (64) ergibt sich hieraus eine Näherung der Matrixexponentialfunktion⁴:

$$e^{-iH\Delta t} \approx \sum_{n=0}^N a_n T_n(H_{\text{norm}}). \quad (79)$$

mit den Entwicklungskoeffizienten aus (65) und den Ersetzungen (75)-(78).

Die Genauigkeit dieses Verfahrens wird nur durch das Verhältnis von N und $\Delta E\Delta t$ bestimmt, sodaß die Schrittweite Δt prinzipiell keinen Beschränkungen unterliegt und beliebig gewählt werden kann, wenn N genügend groß ist.

2.2 Rationale Approximationen

2.2.1 Motivation

In 2.1.1 wurde die Matrixexponentialfunktion durch

$$e^{-iH\Delta t} = 1 - iH\Delta t + O(\Delta t^2) \quad (80)$$

approximiert. Wie dort bereits angemerkt, ist diese Näherung nicht unitär. Besser ist die Approximation der Form

$$e^{-iH\Delta t} = \frac{1 - iH(\Delta t/2)}{1 + iH(\Delta t/2)} + O(\Delta t^3), \quad (81)$$

die unitär und von höherer Ordnung in Δt ist. Gleichung (81) läßt sich für einen zeitabhängigen Hamiltonoperator der Form $H = T + V(t)$ noch zu

$$U(\Delta t) = \frac{1 - iH \cdot (\Delta t/2) - i\dot{V} \cdot (\Delta t^2/4)}{1 + iH \cdot (\Delta t/2) + i\dot{V} \cdot (\Delta t^2/4)} + O(\Delta t^3) \quad (82)$$

⁴Die Tschebyscheff-Entwicklung kann auch für Matrixexponentialfunktionen mit reellem Argument verwendet werden, z. B. für die Berechnung von $e^{-\beta H}$ in Gl. (5). Dies läßt sich analog zu obigen Überlegungen auf $e^{-x\Delta z}$ zurückführen. Auch hier kann nach [17, Gl. (9.6.34)] eine Entwicklung der Form (61) angesetzt werden, wenn die Besselfunktionen J_k durch die *Besselfunktionen mit rein imaginären Variablen* I_k ersetzt werden. Diese sind mit den J_k verknüpft über [12, S. 399, Gl. (9.53a)]

$$I_k(x) = i^{-k} J_k(ix).$$

verallgemeinern [19].

Man sieht, daß diese Verfahren umso genauer sind, je kleiner die Zeitschritte Δt gewählt werden. Dies führt aber zu einem sehr großen Rechenaufwand, da für jeden Zeitschritt u.U. sehr große Matrizen zu invertieren sind. Ist man also nicht an einer hohen zeitlichen Auflösung interessiert, sind Verfahren günstiger, deren Genauigkeit unabhängig von Δt gewählt werden kann. Ein derartiges Verfahren liefert die Padé-Approximation.

2.2.2 Padé-Approximationen

Das Padé-Verfahren beruht auf der Näherung der Exponentialfunktion durch eine rationale Funktion.

Es gilt der Zusammenhang [20]

$$e^{-i\mathbf{H}\Delta t} = [L/M](-i\mathbf{H}\Delta t) + O(\Delta t^{L+M+1}). \quad (83)$$

Die Padé-Approximante $[L/M](z)$ ist definiert durch zwei Polynome $p(z)$ und $q(z)$ vom Grad $\deg p = L$ bzw. $\deg q = M$:

$$[L/M](z) = \frac{p(z)}{q(-z)}, \quad (84)$$

wobei $p(z)$ und $q(z)$ durch [13]

$$p(z) = \sum_{j=0}^L \frac{(L+M-j)! L!}{(L+M)! j! (L-j)!} z^j \quad (85)$$

$$q(z) = \sum_{j=0}^M \frac{(L+M-j)! M!}{(L+M)! j! (M-j)!} z^j \quad (86)$$

gegeben sind.

Man sieht aus (83), daß der Fehler in Δt die Ordnung $L+M+1$ besitzt, und demzufolge die Schrittweite Δt schon bei kleinen Werten für L und M relativ groß gewählt werden kann, verglichen mit anderen Verfahren.

Der Nachteil dieses Verfahrens ist, daß zur Berechnung der Matrixexponentialfunktion die Matrix $q(+i\mathbf{H}\Delta t)$ zu invertieren ist, was sehr rechenintensiv ist und außerdem die Genauigkeit durch Rundungsfehler stark einschränken kann. Besonders in der Nähe von Nullstellen des Nennerpolynoms $q(z)$ können kleine Abweichungen, dividiert durch noch kleinere Werte des Nenners, zu extremen Fehlern führen.

In [13] wird die Padé-Approximation für Matrizen \mathbf{A} empfohlen, deren Norm $\|\mathbf{A}\|$ (siehe (41)) nicht zu groß ist. Eine Verbesserung des Konvergenzverhaltens könnte auch hier durch geeignete Skalierung (Gleichung (47), siehe auch [14]) erreicht werden. [13] und [14] verwenden die diagonale Padé-Approximation mit $L = M$, da bei etwa gleichem Rechenaufwand ($L-1$ Matrixmultiplikationen) die $[L/L]$ -Approximation die geringsten Rundungsfehler und größtmögliche Stabilität aufweist.

In diesem Fall ist wegen $L = M$ auch

$$p(z) = q(z) = \sum_{j=0}^L \frac{(2L-j)! L!}{(2L)! j! (L-j)!} z^j \quad (87)$$

und die Padé-Approximante gegeben durch

$$[L/L](z) = [p(-z)]^{-1} p(z). \quad (88)$$

Gleichung (81) stellt übrigens die diagonale Padé-Approximante mit $L = 1$ dar:

$$e^{-iH\Delta t} = [1/1](-iH\Delta t) + O(\Delta t^3) = \frac{1 - iH(\Delta t/2)}{1 + iH(\Delta t/2)} + O(\Delta t^3). \quad (89)$$

Eine Erweiterung des Padé-Verfahrens wird in [21] vorgeschlagen:

Man fügt zur $[L/L](A)$ -Approximation den Term

$$\frac{cA^{2L+1}}{p(-A)^2} = cA^{2L+1} + cA^{2L+2} + O(A^{2L+3}) \quad (90)$$

hinzu. Ein Vergleich mit der Taylorentwicklung zeigt, daß die Approximation (90) die Ordnung $2L + 2$ besitzt, wenn

$$c = \frac{(-1)^L (L!)^2}{(2L+1)! (2L)!} \quad (91)$$

gewählt wird. Der Vorteil hierbei ist, daß die erzielte Genauigkeit der $[L+1/L+1]$ -Approximation entspricht, obwohl der Rechenaufwand geringer ist als bei der direkten Berechnung dieser Approximation nach Gl. (88).

2.3 Matrixextrapolationsverfahren

Ein weiterer Zugang zur Berechnung der Matrixexponentialfunktion bietet sich durch die Verwendung von *Extrapolationsverfahren*, zu denen strenggenommen auch das im vorigen Abschnitt vorgestellte Padé-Verfahren zählt. Dies wird in Abschnitt 2.3.1 deutlich.

Ausgangspunkt der folgenden Überlegungen ist die Taylorapproximation der Matrixexponentialfunktion in der Form

$$p_N(A) = \sum_{k=0}^N \frac{A^k}{k!}. \quad (92)$$

(siehe (45), Kap. 2.1.1).

Definiert man nun

$$S_N := p_N(A), \quad (93)$$

so erhält man für wachsenden Approximationsgrad N eine Folge von Matrizen

$$S_0, S_1, S_2, \dots, \quad (94)$$

die gegen $S_\infty = \exp(\mathbf{A})$ konvergiert. Zum Konvergenzverhalten der Taylorreihe siehe auch Kapitel 2.1.1.

Um eine Verbesserung des Konvergenzverhaltens zu erhalten, bietet sich eine Folgentransformation an, die die Folge (94) in eine transformierte Folge

$$\tilde{S}_0, \tilde{S}_1, \tilde{S}_2 \dots \quad (95)$$

überführt. Konvergiert (95) schneller als (94), war die Transformation erfolgreich [22].

Im folgenden werden einige Transformations-Algorithmen vorgestellt.

2.3.1 ϵ -Algorithmus und verwandte Verfahren

Der ϵ -Algorithmus ist durch das Rekursionsschema

$$\begin{aligned} \epsilon_{-1}^{(n)} &= 0, & \epsilon_0^{(n)} &= s_n, \\ \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + \frac{1}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}} \end{aligned} \quad (96)$$

definiert [23]. Die s_n bilden die skalare Ausgangsreihe. Dabei sind die $\epsilon_k^{(n)}$ mit ungeradem k nur Hilfsgrößen, während die mit geradem k die transformierte Folge darstellen. Dieser Algorithmus kann direkt in ein Verfahren für quadratische Matrizen übersetzt werden gemäß

$$\begin{aligned} \epsilon_{-1}^{(n)} &= 0, & \epsilon_0^{(n)} &= S_n, \\ \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + \left[\epsilon_k^{(n+1)} - \epsilon_k^{(n)} \right]^{-1}. \end{aligned} \quad (97)$$

Man sieht, daß in jedem Iterationsschritt Matrixinverse zu berechnen sind, was die Anwendung dieses Verfahrens hauptsächlich auf kleine Matrizen einschränkt.

Zu beachten ist, daß der ϵ -Algorithmus die obere Hälfte der Padé-Tafel berechnet. Es gilt der Zusammenhang [23, 24]

$$\epsilon_{2k}^{(n)} = [n + k/k]_f(z), \quad k \geq 0, n \geq 0, \quad (98)$$

wenn für die s_n die Partialsummen

$$f_n(z) = \sum_{k=0}^n c_k z^k \quad (99)$$

der Potenzreihenentwicklung von $f(z)$ eingesetzt werden.

2.3.2 Levin-artige Verfahren

Der Prototyp aller Levin-artigen Transformationen ist die *Levin-Transformation* [25]:

$$\mathcal{L}_k^{(n)}(\beta, s_n, \omega_n) = \frac{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} \frac{s_{n+j}}{\omega_{n+j}}}{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} \frac{1}{\omega_{n+j}}} \quad (100)$$

β ist hierbei eine Konstante, die ω_n sind sogenannte *Restabschätzungen*, d. h. Größen, die die Beziehung

$$\lim_{n \rightarrow \infty} \frac{s - s_n}{\omega_n} = c, \quad 0 < |c| < \infty \quad (101)$$

zumindest näherungsweise erfüllen [22]. s bezeichnet den Limes der Folge $\{s_n\}$. Man hat also eine Folgentransformation der Form

$$\{s_0, s_1, \dots, s_k\} \longrightarrow \{\mathcal{L}_0^{(0)}, \mathcal{L}_1^{(0)}, \dots, \mathcal{L}_k^{(0)}\}. \quad (102)$$

In der Praxis sind folgende Restabschätzungen und Bezeichnungen üblich:

$$t\text{-Variante: } {}^t\omega_n = \Delta s_{n-1} \quad (103)$$

$$\tilde{t}\text{-Variante: } \tilde{t}\omega_n = \Delta s_n \quad (104)$$

$$u\text{-Variante: } {}^u\omega_n = (n + \beta)\Delta s_{n-1} \quad (105)$$

$$v\text{-Variante: } {}^v\omega_n = -\Delta s_n \Delta s_{n-1} / \Delta^2 s_{n-1} \quad (106)$$

Der Differenzenoperator Δ wirkt hier gemäß Definition (36) bzw. (38). Obige Beziehungen lassen sich ebenso für Matrizen und – abgesehen von der v -Variante nach (106) – auch für Vektoren verwenden.

Sind die s_n als Partialsummen einer Potenzreihe in z darstellbar, also von der Form

$$s_n = \sum_{j=0}^n a_j z^j \quad (107)$$

mit Koeffizienten a_j , so ergibt sich für die Restabschätzung im Falle der t -Variante:

$$\omega_n = a_n z^n. \quad (108)$$

Setzt man (107) und (108) in (100) ein, erhält man

$$\mathcal{L}_k^{(n)}(\beta, s_n, \omega_n) = \frac{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} \frac{P_{n+j}(z)}{a_{n+j} z^{n+j}}}{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} \frac{1}{a_{n+j} z^{n+j}}}. \quad (109)$$

Werden Zähler und Nenner mit z^{n+k} durchmultipliziert, erhält man jeweils einen polynomialen Ausdruck in z , insgesamt ist $\mathcal{L}_k^{(n)}(\beta, s_n, \omega_n)$ also eine rationale Funktion in z und damit mit den in Abschnitt 2.2 vorgestellten Verfahren verwandt.

Auch die skalare Levin-Transformation läßt sich auf Matrizen übertragen. Die S_n sind wegen (93) von der Form (107), also Polynome vom Grad n mit dem Argument A und können daher in (109) eingesetzt werden. Für $A = -iH\Delta t$ resultiert daraus eine rationale Approximation des Zeitentwicklungsoperators.

Aus (100) folgt die Matrix-Levin-Transformation⁵

$$\mathcal{L}_k^{(n)}(\beta, S_n, \omega_n) = \frac{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} S_{n+j} \omega_{n+j}^{-1}}{\sum_{j=0}^k (-1)^j \binom{k}{j} \frac{(\beta + n + j)^{k-1}}{(\beta + n + k)^{k-1}} \omega_{n+j}^{-1}}. \quad (110)$$

Durch die bereits erwähnte Erweiterung von Zähler und Nenner mit z^{n+k} bei festgelegtem n und k kann die Invertierung der ω_{n+j} umgangen werden, sodaß insgesamt nur *eine* Matrixinversion zur Berechnung von $\mathcal{L}_k^{(n)}(\beta, s_n, \omega_n)$ notwendig wird.

2.4 Vektorextrapolationsverfahren

Prinzipiell kann man jede $(D \times D)$ -Matrix A durch Umsortieren der Matrixelemente in einen D^2 -dimensionalen Vektor \vec{A} überführen:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1D} \\ \vdots & \ddots & \vdots \\ a_{D1} & \cdots & a_{DD} \end{pmatrix} \longrightarrow \vec{A} = \begin{pmatrix} a_{11} \\ \vdots \\ a_{D1} \\ \vdots \\ a_{1D} \\ \vdots \\ a_{DD} \end{pmatrix}. \quad (111)$$

Wendet man (111) auf die Matrizen der Folge (94) der Taylormatrizen an, so erhält man eine Folge von Vektoren

$$\vec{S}_0, \vec{S}_1, \dots, \quad (112)$$

auf die man Vektorextrapolationsverfahren anwenden kann, um den Vektor \vec{S}_∞ zu bestimmen, aus dem man schließlich die Matrix $S_\infty = \exp(A)$ durch Umordnung der Elemente in Umkehrung zu (111) erhält.

Da für Vektoren $\vec{v} \neq 0$ kein Inverses existiert, können skalare Algorithmen nicht ohne weiteres auf Vektoren übertragen werden. Immerhin ist es jedoch möglich, *Pseudoinverse* zu definieren, um dieses Problem zu umgehen und skalare Methoden auf Vektorverfahren zu erweitern.

⁵Wenn S_{n+j} und ω_{n+j} nicht vertauschen, existieren zwei verschiedene Transformationen, je nachdem, ob im Zähler $S_{n+j} \omega_{n+j}^{-1}$ oder $\omega_{n+j}^{-1} S_{n+j}$ gesetzt wird. Mit der vorliegenden speziellen Wahl von S_n und ω_n sind beide Transformationen jedoch identisch.

Eine mögliche Definition des Pseudoinversen ist durch

$$\vec{v}^- := \frac{\vec{v}}{(\vec{v}, \vec{v})} \quad (113)$$

mit dem Skalarprodukt

$$(\vec{a}, \vec{b}) = (\vec{a}^*)^T \vec{b} = \vec{a}^\dagger \vec{b} \quad (114)$$

gegeben.

Einige Vektorextrapolationsverfahren werden im folgenden erklärt.

2.4.1 ϵ -Algorithmen

Mit der Definition (113) kann der ϵ -Algorithmus (96) aus Abschnitt 2.3.1 direkt in einen *vektoriellen ϵ -Algorithmus* übersetzt werden [26, Kap. 4.1]:

$$\begin{aligned} \vec{\epsilon}_{-1}^{(n)} &= 0, & \vec{\epsilon}_0^{(n)} &= \vec{S}_n, \\ \vec{\epsilon}_{k+1}^{(n)} &= \vec{\epsilon}_{k-1}^{(n+1)} + [\vec{\epsilon}_k^{(n+1)} - \vec{\epsilon}_k^{(n)}]^- . \end{aligned} \quad (115)$$

Durch die Verwendung des Pseudoinversen ist dieser Algorithmus auch für größere Matrizen geeignet, vgl. Kap. 2.3.1.

Eine andere Variation des ϵ -Algorithmus stellt der *topologische ϵ -Algorithmus* dar [26, Kap. 4.2]. Davon existieren zwei Varianten: der *erste topologische ϵ -Algorithmus* in der Form

$$\begin{aligned} \vec{\epsilon}_{-1}^{(n)} &= 0, & \vec{\epsilon}_0^{(n)} &= \vec{S}_n, \\ \vec{\epsilon}_{2k+1}^{(n)} &= \vec{\epsilon}_{2k-1}^{(n+1)} + \frac{\vec{y}}{\vec{y}^T (\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)})}, \\ \vec{\epsilon}_{2k+2}^{(n)} &= \vec{\epsilon}_{2k}^{(n+1)} + \frac{\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)}}{(\vec{\epsilon}_{2k+1}^{(n+1)} - \vec{\epsilon}_{2k+1}^{(n)})^T (\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)})}, \end{aligned} \quad (116)$$

sowie der *zweite topologische ϵ -Algorithmus*, der sich vom ersten nur durch die Rekursionsvorschrift für gerade untere Indizes unterscheidet. Man erhält ihn, indem man die letzte Zeile von Gl. (116) durch

$$\vec{\epsilon}_{2k+2}^{(n)} = \vec{\epsilon}_{2k}^{(n+1)} + \frac{\vec{\epsilon}_{2k}^{(n+2)} - \vec{\epsilon}_{2k}^{(n+1)}}{(\vec{\epsilon}_{2k+1}^{(n+1)} - \vec{\epsilon}_{2k+1}^{(n)})^T (\vec{\epsilon}_{2k}^{(n+2)} - \vec{\epsilon}_{2k}^{(n+1)})}. \quad (117)$$

ersetzt.

Der Vektor $\vec{y} \neq 0$ in (116) ist hierbei beliebig, aber fest gewählt. Auch bei den Algorithmen (115), (116) und (117) sind die Größen $\vec{\epsilon}_{2k+1}^{(n)}$, also mit ungeradem unteren Index, nur Hilfsgrößen.

2.4.2 Projektionsmethoden

Eine große Klasse von Vektorextrapolationsverfahren stellen die *Projektionsmethoden* dar. Diese Bezeichnung rührt daher, daß versucht wird, die Abweichungen einer Vektorfolge vom Grenzwert aus dieser Folge herauszuprojizieren, um so das Konvergenzverhalten zu beschleunigen.

Die ϵ -Algorithmen, die bereits in Abschnitt 2.4.1 behandelt wurden, bilden eine wichtige Gruppe von Projektionsalgorithmen. Die zweite große Gruppe beinhaltet die *polynomialen Verfahren*. Hierzu gehören das *minimal polynomial extrapolation*-Verfahren (MPE) [27], das modifizierte MPE-Verfahren (*modified minimal polynomial extrapolation method*, MMPE) [28] sowie die *reduced rank extrapolation*-Methode [29, 30].

Diese Verfahren lassen sich in einer Formel zusammenfassend darstellen: Die Vektoren der Folge seien mit \vec{S}_n bezeichnet. Ihre Dimension sei D . $\Delta_{n,k}^j$ ist die $(D \times k)$ -Matrix mit den $\Delta^j \vec{S}_n, \dots, \Delta^j \vec{S}_{n+k-1}$ als Spaltenvektoren⁶. Aus einer gegebenen Hilfsfolge von D -dimensionalen Vektoren $\vec{y}_k^{(n)}$, die frei bestimmbar sind, bildet man die $(D \times k)$ -Matrix $Y_{n,k}$ mit den Vektoren $\vec{y}_1^{(n)}, \dots, \vec{y}_k^{(n)}$.

Nun wird eine Transformation $\{\vec{S}_n\} \rightarrow \{\vec{T}_{n,k}\}$ definiert durch [26]:

$$\vec{T}_{n,k} = \vec{S}_n - \Delta_{n,k}^1 \left(Y_{n,k}^\dagger \Delta_{n,k}^2 \right)^{-1} Y_{n,k}^\dagger \Delta \vec{S}_n \quad (118)$$

Für $\vec{y}_j^{(n)} = \Delta \vec{S}_{n+j+1}$ erhält man das MPE-Verfahren. Setzt man $\vec{y}_j^{(n)} = \vec{y}_j$ mit festen Vektoren \vec{y}_j , so hat man das MMPE-Verfahren und für $\vec{y}_j^{(n)} = \Delta^2 \vec{S}_{n+j+1}$ die RRE-Methode.

2.4.3 Vektor- \mathcal{J} -Transformation

Die *Vektor- \mathcal{J} -Transformation* ist gegeben durch [22]

$$\begin{aligned} \vec{s}_n^{(0)} &= \vec{S}_n, & \vec{\omega}_n^{(0)} &= \vec{\omega}_n, \\ \vec{s}_n^{(k+1)} &= \vec{s}_{n+1}^{(k)} - \frac{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{s}_n^{(k)}}{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\ \vec{\omega}_n^{(k+1)} &= \mathbf{r}_{n+1}^{(k)} \vec{\omega}_{n+1}^{(k)} - \frac{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta [\mathbf{r}_n^{(k)} \vec{\omega}_n^{(k)}]}{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\ {}_V \vec{\mathcal{J}}_n^{(k)} \left(\{\vec{S}_n\}, \{\vec{\omega}_n\}, \{\mathbf{r}_n^{(k)}\} \right) &= \vec{s}_n^{(k)} \end{aligned} \quad (119)$$

mit quadratischen Matrizen $\mathbf{r}_n^{(k)}$ (man beachte die vertauschte Anordnung der Indizes n und k). Statt der $\mathbf{r}_n^{(k)}$ können auch Vektoren $\vec{u}_n^{(k)} = \mathbf{r}_n^{(k)} \vec{\omega}_n^{(k)}$ vorgegeben werden, was

⁶Zur Definition der Differenzenoperatoren Δ^j siehe (36) und (37).

den Speicherbedarf reduziert und auf folgenden Algorithmus führt:

$$\begin{aligned}
\vec{s}_n^{(0)} &= \vec{S}_n, & \vec{\omega}_n^{(0)} &= \vec{\omega}_n, \\
\vec{s}_n^{(k+1)} &= \vec{s}_{n+1}^{(k)} - \frac{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{s}_n^{(k)}}{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\
\vec{\omega}_n^{(k+1)} &= \vec{u}_{n+1}^{(k)} - \frac{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{u}_n^{(k)}}{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\
{}_V \vec{\mathcal{J}}_n^{(k)} \left(\{\vec{S}_n\}, \{\vec{\omega}_n\}, \{r_n^{(k)}\} \right) &= \vec{s}_n^{(k)}.
\end{aligned} \tag{120}$$

Setzt man die Matrizen $r_n^{(k)}$ als Vielfache der Einheitsmatrix $\mathbf{1}$, also $r_n^{(k)} = r_n^{(k)} \mathbf{1}$, so folgt $\vec{u}_n^{(k)} = r_n^{(k)} \vec{\omega}_n^{(k)}$ und damit

$$\begin{aligned}
\vec{s}_n^{(0)} &= \vec{S}_n, & \vec{\omega}_n^{(0)} &= \vec{\omega}_n, \\
\vec{s}_n^{(k+1)} &= \vec{s}_{n+1}^{(k)} - \frac{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{s}_n^{(k)}}{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\
\vec{\omega}_n^{(k+1)} &= - \frac{[\Delta\vec{\omega}_n^{(k)}]^\dagger \vec{\omega}_n^{(k)}}{[\Delta\vec{\omega}_n^{(k)}]^\dagger \Delta\vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)} \Delta r_n^{(k)}, \\
{}_V \vec{\mathcal{J}}_n^{(k)} \left(\{\vec{S}_n\}, \{\vec{\omega}_n\}, \{r_n^{(k)} \mathbf{1}\} \right) &= \vec{s}_n^{(k)}.
\end{aligned} \tag{121}$$

Ähnlich wie beim ϵ -Algorithmus existiert auch bei der \mathcal{J} -Transformation eine topologische Variante, die *topologische \mathcal{J} -Transformation*. Man erhält sie aus den Rekursionen (119)-(121), indem man jeweils $[\Delta\vec{\omega}_n^{(k)}]^\dagger$ durch $[\vec{y}_n^{(k)}]^T$ ersetzt. Die Vektoren $\vec{y}_n^{(k)} \neq 0$ sind wieder frei wählbar.

Für die Wahl der Restabschätzungen $\vec{\omega}_n$ gelten die Anmerkungen in Abschnitt 2.3.2. Auch im Falle der \mathcal{J} -Transformation sind gängige Restabschätzungen durch Gln. (103)-(106) gegeben.

2.5 Weitere Verfahren

2.5.1 Diagonalisierung

Das naheliegendste Verfahren zur Berechnung der Matrixexponentialfunktion stellt die *Diagonalisierung* dar. Gesucht wird also eine Matrix X , die A auf Diagonalgestalt

bringt, d. h. die

$$D = X^{-1} A X, \quad D = \begin{pmatrix} d_0 & & & 0 \\ & d_1 & & \\ & & \ddots & \\ 0 & & & d_D \end{pmatrix} \quad (122)$$

erfüllt. Gleichbedeutend mit der Bestimmung von X ist die Berechnung aller Eigenwerte von A , die ja die Einträge d_1, d_2, \dots, d_D in der Hauptdiagonalen von D bilden. Ist D bekannt, kann die Matrixexponentialfunktion berechnet werden:

$$e^A = X e^D X^{-1} = X \begin{pmatrix} e^{d_0} & & & 0 \\ & e^{d_1} & & \\ & & \ddots & \\ 0 & & & e^{d_D} \end{pmatrix} X^{-1}. \quad (123)$$

Dieses Verfahren ist sehr rechenintensiv, da X berechnet und invertiert werden muß und außerdem sehr anfällig gegenüber Rundungsfehlern, da Fehler in den d_j ($j = 0, \dots, D$) bei der Berechnung von e^{d_j} exponentiell anwachsen.

2.5.2 Programme zur Lösung von Differentialgleichungen

In [13] wird darauf hingewiesen, daß bereits bestehende Computerprogramme zur Lösung von Differentialgleichungen auch zur Berechnung von $\exp(At)$ eingesetzt werden können.

Diese Programme sind im allgemeinen darauf ausgelegt, ein System der Form

$$\dot{x} = f(x, t), \quad x(0) = x_0 \quad (124)$$

zu lösen. Wählt man $f(x, t) = Ax$, so erhält man die k -te Spalte von $\exp(At)$, wenn als Randbedingung x_0 gleich der k -ten Spalte der Einheitsmatrix gesetzt wird.

Für größere Matrizen ist diese Methode sehr ineffektiv, da diese Programme die explizite Zeitunabhängigkeit der auftretenden Koeffizienten in A i. a. nicht ausnutzen [13].

2.5.3 Cauchysche Integralformel

Für Funktionen $f(z)$, die auf einer geschlossenen Kurve K und in dem von ihr umschlossenen Gebiet analytisch sind, gilt der *Cauchysche Integralsatz*:

$$f(z) = \frac{1}{2\pi i} \oint_{(K)} \frac{f(\xi)}{\xi - z} d\xi, \quad (125)$$

wenn z in diesem Gebiet liegt [12]. ξ durchläuft dabei die Kurve K im Gegenuhrzeigersinn. Dies kann zur Berechnung der Matrixexponentialfunktion herangezogen werden.

Setzt man für die Funktion $f(z)$ die Exponentialfunktion mit dem Argument A ein, so erhält man

$$e^A = \frac{1}{2\pi i} \oint_{(K_{\sigma(A)})} \frac{e^\xi}{\xi 1 - A} d\xi. \quad (126)$$

Die Kurve $K_{\sigma(A)}$ schließt dabei das gesamte Spektrum der Eigenwerte von A ein. Auch hier muß also der Bereich, in dem die Eigenwerte liegen, bekannt sein bzw. abgeschätzt werden (vgl. hierzu Kap. 2.1.3). Darüberhinaus muß die Matrix $\xi 1 - A$ für alle ξ invertiert werden, was insbesondere bei sehr großen Matrizen zu einem erheblichen Rechenaufwand führt. Falls man das Integral numerisch auswertet, wobei der Integrand an endlich vielen Stellen ξ_n , $n = 1, \dots, N$ zur Berechnung herangezogen wird, müssen N Matrixinversionen durchgeführt werden.

3 Verfahren zur Propagation von Wellenpaketen

Ist man nur am Endzustand $\psi(t)$ interessiert, kann die explizite Berechnung der Matrix $\exp(-i\mathbf{H}t)$ umgangen werden. Die im folgenden dargestellten Verfahren stellen größtenteils Modifikationen der im Abschnitt 2 vorgestellten Methoden dar, sodaß auf eine Wiederholung der Herleitungen verzichtet und nur die Unterschiede zu den Matrizenverfahren herausgearbeitet wurden.

Eine Ausnahme bildet das *split operator*-Verfahren, das zwar eine Umformung der Matrixexponentialfunktion beinhaltet, aber durch seine Verbindung mit schneller Fourier-Transformation eher zu den Propagationsverfahren zu zählen ist. Eine Beschreibung erfolgt in Abschnitt 3.4.

3.1 Polynomiale Approximationen

3.1.1 Taylor-Reihe

Ausgangspunkt ist die Taylorentwicklung der Matrixexponentialfunktion nach Gleichung (45) (vgl. Kap. 2.1.1) mit $-i\mathbf{H}t$ als Matrixargument, angewandt auf die Wellenfunktion zum Zeitpunkt $t_0 = 0$:

$$\psi(t) = e^{-i\mathbf{H}t}\psi(0) \approx p_N(-i\mathbf{H}t)\psi(0). \quad (127)$$

Durchläuft der Approximationsgrad N die Werte $n = 0, 1, \dots, \infty$, so erhält man mit

$$\psi_n := p_n(-i\mathbf{H}t)\psi(0) = \sum_{j=0}^n \frac{1}{j!} (-i\mathbf{H}t)^j \psi(0) \quad (128)$$

eine Folge von Vektoren ψ_0, ψ_1, \dots , die für $n \rightarrow \infty$ gegen $\psi(t)$ strebt.

Diese Vektorfolge kann rekursiv berechnet werden: aus der Definition (128) folgt für den Vektor ψ_{n+1} die Gleichung

$$\begin{aligned} \psi_{n+1} &= \sum_{j=0}^{n+1} \frac{1}{j!} (-i\mathbf{H}t)^j \psi(0) = \psi_n + \frac{1}{(n+1)!} (-i\mathbf{H}t)^{n+1} \psi(0) \\ &= \psi_n - \frac{i\mathbf{H}t}{n+1} \Delta\psi_{n-1}, \end{aligned} \quad (129)$$

wobei $\Delta\psi_{n-1}$ nach Gleichung (36) definiert wurde als

$$\Delta\psi_{n-1} := \psi_n - \psi_{n-1} = \frac{1}{n!} (-i\mathbf{H}t)^n \psi(0). \quad (130)$$

Zusammenfassend gilt also die Rekursion

$$\begin{aligned} \psi_0 &= \psi(0), & \psi_1 &= (-i\mathbf{H}t)\psi(0) \\ \psi_{n+1} &= \psi_n - \frac{i\mathbf{H}t}{n+1} \Delta\psi_{n-1} & (n \geq 1). \end{aligned} \quad (131)$$

In der Praxis kann die Folge von Vektoren so lange berechnet werden, bis im Rahmen der gewünschten oder erreichbaren Genauigkeit $\psi_{n+1} = \psi_n$ gilt. Der Vorteil dieses Verfahrens liegt im verringerten Rechenaufwand gegenüber der Taylorentwicklung nach Kap. 2.1.1, da keine Matrixmultiplikationen durchzuführen sind und nur die Matrix (iHt) abgespeichert sein muß.

3.1.2 Cayley-Hamilton-Verfahren

Auch hier lassen sich Rechenaufwand und Speicherbedarf erheblich reduzieren, wenn man die Matrizen A^0, A^1, \dots, A^{n-1} (siehe Kap. 2.1.2) jeweils mit dem Zustandsvektor $\psi(0)$ multipliziert. Man erhält dann eine Folge von Zustandsvektoren $\psi_0 = \psi(0), \psi_1 = A\psi_0, \dots, \psi_{n-1} = A\psi_{n-2}$, die in die Entwicklung (52) eingesetzt werden können. Daraus ergibt sich

$$e^{At}\psi(0) = \sum_{j=0}^{n-1} \alpha_j(t)\psi_j, \quad A = -iH. \quad (132)$$

Zur Definition der $\alpha_j(t)$ siehe Kap. 2.1.2.

3.1.3 Tschebyscheff-Methode

Wie bereits in Kap. 2.1.3 hergeleitet, läßt sich die Matrixexponentialfunktion nach den Tschebyscheff-Polynomen T_n entwickeln:

$$e^{-iHt} = \sum_{n=0}^{\infty} a_n T_n(H_{\text{norm}}). \quad (133)$$

Die Größen a_n und H_{norm} sind in (65) bzw. (72) definiert. Angewandt auf die Wellenfunktion ergibt sich hieraus

$$\psi(t) = e^{-iHt}\psi(0) = \sum_{n=0}^{\infty} a_n T_n(H_{\text{norm}})\psi(0) \quad (134)$$

$$= \sum_{n=0}^{\infty} a_n \psi_n \quad (135)$$

mit der Substitution

$$\psi_n = T_n(H_{\text{norm}})\psi(0). \quad (136)$$

Zur rekursiven Berechnung der ψ_n kann Gleichung (63) verwendet werden; aus ihr folgt

$$\psi_0 = \psi(0), \quad \psi_1 = H_{\text{norm}}\psi(0) \quad (137)$$

$$\psi_{n+1} = 2H_{\text{norm}}\psi_n - \psi_{n-1} \quad (n \geq 1).$$

3.2 Extrapolationsverfahren mit Matrixinversion

Grundsätzlich lassen sich die in den Abschnitten 2.2 und 2.3 beschriebenen Verfahren auch zur Propagation des Zustandsvektors verwenden, indem man die aus ihnen erhaltenen Näherungen direkt auf den Zustandsvektor anwendet. Das Padé-Verfahren nach Kap. 2.2.2 wird hier durch seine Verbindung mit dem ϵ -Algorithmus (siehe auch die Ausführungen unter Kap. 2.3.1) als Extrapolationsmethode behandelt.

Nachdem aber ein schwerwiegender Nachteil dieser Verfahren – nämlich die Durchführung einer oder mehrerer Matrixinversionen pro Extrapolationsschritt – auch durch den Übergang auf den Zustandsvektor nicht beseitigt wird, sind diese allenfalls von theoretischem Interesse und sollen hier deshalb nicht weiter verfolgt werden.

3.3 Extrapolationsverfahren ohne Matrixinversion

Der erhebliche Rechenaufwand, den eine Matrixinversion mit sich bringt, wird durch die Verwendung der im folgenden beschriebenen Methoden umgangen. Die Anwendung von Vektorextrapolationsmethoden auf eine Folge von Zustandsvektoren ist aber noch aus anderer Sicht interessant:

Durch die Umformung einer $(D \times D)$ -Matrix zu einem Vektor nach (111) erzeugt einen D^2 -dimensionalen Spaltenvektor. Der Zustandsvektor zu einem Problem mit einer $(D \times D)$ -Hamiltonmatrix besitzt aber nur die Dimension D , somit verringert sich der Rechenaufwand im Vergleich zu den in Kap. 2.4 vorgestellten Methoden ganz erheblich.

3.3.1 ϵ -Algorithmen

Der Vektor- ϵ -Algorithmus nach (115) kann direkt auf die Folge ψ_0, ψ_1, \dots der Wellenfunktionen, definiert durch (128), angewandt werden:

$$\begin{aligned} \vec{\epsilon}_{-1}^{(n)} &= 0, & \vec{\epsilon}_0^{(n)} &= \psi_n, \\ \vec{\epsilon}_{k+1}^{(n)} &= \vec{\epsilon}_{k-1}^{(n+1)} + \left[\vec{\epsilon}_k^{(n+1)} - \vec{\epsilon}_k^{(n)} \right]^{-}. \end{aligned} \tag{138}$$

Alternativ dazu sind auch die topologischen ϵ -Algorithmen möglich:

$$\begin{aligned}\vec{\epsilon}_{-1}^{(n)} &= 0, & \vec{\epsilon}_0^{(n)} &= \psi_n, \\ \vec{\epsilon}_{2k+1}^{(n)} &= \vec{\epsilon}_{2k-1}^{(n+1)} + \frac{\vec{y}}{\vec{y}^T(\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)})}, \\ \vec{\epsilon}_{2k+2}^{(n)} &= \vec{\epsilon}_{2k}^{(n+1)} + \frac{\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)}}{(\vec{\epsilon}_{2k+1}^{(n+1)} - \vec{\epsilon}_{2k+1}^{(n)})^T(\vec{\epsilon}_{2k}^{(n+1)} - \vec{\epsilon}_{2k}^{(n)})}\end{aligned}\quad (139)$$

bzw.

$$\vec{\epsilon}_{2k+2}^{(n)} = \vec{\epsilon}_{2k}^{(n+1)} + \frac{\vec{\epsilon}_{2k}^{(n+2)} - \vec{\epsilon}_{2k}^{(n+1)}}{(\vec{\epsilon}_{2k+1}^{(n+1)} - \vec{\epsilon}_{2k+1}^{(n)})^T(\vec{\epsilon}_{2k}^{(n+2)} - \vec{\epsilon}_{2k}^{(n+1)})}.$$

Auch hier ist \vec{y} ein Hilfsvektor, siehe Abschnitt 2.3.1.

3.3.2 Projektionsmethoden

Die in Abschnitt 2.4.2 vorgestellten Projektionsalgorithmen können direkt auf die Wellenfunktionen angewandt werden, wenn man in der Definition der Matrix $\Delta_{n,k}^j$ die Vektoren \vec{S}_n durch die ψ_n aus Gleichung (128) ersetzt.

$$\vec{T}_{n,k} = \psi_n - \Delta_{n,k}^1 \left(\mathbf{Y}_{n,k}^\dagger \Delta_{n,k}^2 \right)^{-1} \mathbf{Y}_{n,k}^\dagger \Delta \psi_n \quad (140)$$

3.3.3 Vektor- \mathcal{J} -Transformation

Sowohl die Vektor- \mathcal{J} -Transformation als auch die topologische \mathcal{J} -Transformation lassen sich zur Extrapolation der Wellenfunktion verwenden, wenn man in den Gleichungen (119), (120) und (121) die Ersetzung $\vec{S}_n \rightarrow \psi_n$ vornimmt (zur topologischen \mathcal{J} -Transformation beachte man die Anmerkungen am Ende von Kap. 2.4.3). Exemplarisch soll dies hier für Gleichung (119) geschehen:

$$\begin{aligned}\vec{s}_n^{(0)} &= \psi_n, & \vec{\omega}_n^{(0)} &= \vec{\omega}_n, \\ \vec{s}_n^{(k+1)} &= \vec{s}_{n+1}^{(k)} - \frac{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{s}_n^{(k)}}{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\ \vec{\omega}_n^{(k+1)} &= \mathbf{r}_{n+1}^{(k)} \vec{\omega}_{n+1}^{(k)} - \frac{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta [\mathbf{r}_n^{(k)} \vec{\omega}_n^{(k)}]}{[\Delta \vec{\omega}_n^{(k)}]^\dagger \Delta \vec{\omega}_n^{(k)}} \vec{\omega}_{n+1}^{(k)}, \\ \mathcal{V} \vec{\mathcal{J}}_n^{(k)} \left(\{ \psi_n \}, \{ \vec{\omega}_n \}, \{ \mathbf{r}_n^{(k)} \} \right) &= \vec{s}_n^{(k)}.\end{aligned}\quad (141)$$

3.4 Split-Operator-Verfahren

Das *split-operator* (SO) Verfahren [31, 32] läßt sich anwenden, wenn man den Hamiltonoperator in einen kinetischen und einen potentiellen Anteil aufspalten kann: $\hat{H} = \hat{T} + \hat{V}$. Dann kann die Matrix des Zeitentwicklungsoperators durch symmetrische Aufspaltung des kinetischen Energieoperators folgendermaßen approximiert werden:

$$e^{-iH\Delta t} = e^{-i(T/2)\Delta t} e^{-iV\Delta t} e^{-i(T/2)\Delta t} + O(\Delta t^3). \quad (142)$$

Der Fehler dritter Ordnung in Δt ist eine Folge der Nichtvertauschbarkeit von T und V ⁷.

Das SO-Verfahren macht nun Gebrauch von der Tatsache, daß T im Impulsraum diagonal ist und dort im wesentlichen einer Multiplikation mit dem Quadrat des Wellenvektors k^2 entspricht, während V im Ortsraum Diagonalgestalt aufweist. Die Darstellungen der Wellenfunktion in den beiden Räumen können mit schneller Fouriertransformation (FFT) ineinander umgerechnet werden.

Die Anwendung des Zeitentwicklungsoperators beinhaltet dann folgende Schritte pro Zeitschritt:

1. Transformation der Wellenfunktion in den Impulsraum
2. Multiplikation mit dem Propagator $e^{-i(k^2/4m)\Delta t}$ eines freien Teilchens
3. Rücktransformation in den Ortsraum
4. Multiplikation mit $e^{-iV\Delta t}$
5. Transformation in den Impulsraum
6. Multiplikation mit $e^{-i(k^2/4m)\Delta t}$
7. Rücktransformation in den Ortsraum

Ist man nicht an dem Zustand der propagierten Wellenfunktion nach jedem Zeitschritt interessiert, so kann man mit Ausnahme des ersten und des letzten Schritts die Anwendungen des Freie-Teilchen-Propagators auf die Wellenfunktion zusammenfassen (Schritte 1 und 7 entfallen dann, Schritte 2 und 6 werden durch eine Multiplikation ersetzt), sodaß pro Zeitschritt nur noch eine Transformation und eine Rücktransformation durchgeführt werden muß.

Für die Wahl des Zeitschrittes Δt sollte das Energiespektrum des Hamiltonoperators herangezogen werden. Eine einfache Abschätzung für die Größe von Δt liefert die Heisenbergsche Unschärferelation. Aus ihr folgt als kritischer Wert

$$\Delta t_{\text{krit}} = \frac{1}{E_{\text{max}}}, \quad (143)$$

⁷Dies wird deutlich aufgrund der Beziehung

$$e^{A+B} = \lim_{n \rightarrow \infty} \left(e^{A/n} e^{B/n} \right)^n.$$

den Δt keinesfalls überschreiten sollte, damit sichergestellt ist, daß der Algorithmus stabil bleibt. In der Praxis hat sich

$$\Delta t \approx \frac{\Delta t_{\text{krit}}}{5} \quad (144)$$

gut bewährt [7].

Die split operator-Methode läßt sich erweitern auf eine Genauigkeit bis zur dritten Ordnung in Δt . Es gilt:

$$\begin{aligned} e^{-i\mathbf{H}\Delta t} &= e^{-i\gamma(\mathbf{T}/2)\Delta t} e^{-i\gamma\mathbf{V}\Delta t} \times \\ &\times e^{-i(1-\gamma)(\mathbf{T}/2)\Delta t} e^{-i(1-2\gamma)\mathbf{V}\Delta t} e^{-i(1-\gamma)(\mathbf{T}/2)\Delta t} \times \\ &\times e^{-i\gamma\mathbf{V}\Delta t} e^{-i\gamma(\mathbf{T}/2)\Delta t} + O(\Delta t^4), \end{aligned} \quad (145)$$

mit der Größe

$$\gamma = \frac{1}{2 - \sqrt[3]{2}} \approx 1.351207192\dots \quad (146)$$

Eine Herleitung von (145) findet sich in [33]. Hier können aufgrund der höheren Genauigkeit die Zeitschritte Δt größer gewählt werden, sodaß dieses Verfahren trotz der Tatsache, daß sieben Exponentialoperatoren pro Zeitschritt auf die Wellenfunktion angewandt werden müssen, u. U. effizienter ist als Gleichung (142). Weiterführende Untersuchungen (insbesondere auch zu zeitabhängigen Potentialen und höheren Ordnungen) wurden in [33, 34, 35] durchgeführt.

Eine interessante Variante des SO-Verfahrens bietet sich in Verbindung mit dem Wechselwirkungs-(WW)-Bild. Anders als im Schrödingerbild (im folgenden mit S indiziert), in dem die Wellenfunktion sowohl die dynamische als auch die kinematische Information trägt, beschränkt sich der Informationsgehalt der Wellenfunktion im Wechselwirkungsbild auf den dynamischen Aspekt des Problems. Die Wellenfunktion in der WW-Darstellung stellt demzufolge keine laufende Welle dar, die sich mit fortschreitender Zeit im Raum ausbreitet. Dies hat zwei Vorteile: zum einen muß bei der Festlegung der Größe des Raumgitters, auf dem die numerischen Berechnungen durchgeführt werden, ein Auseinanderlaufen des Wellenpakets nicht mehr berücksichtigt werden, zum anderen können meist größere Zeitschritte gewählt werden, da die Wechselwirkung i. a. nur einen kleinen Teil der Gesamtenergie ausmacht und mit

$$E_{\text{WW}} \ll E_{\text{tot}} \quad (147)$$

und

$$\omega_i = E_i, \quad i = \begin{cases} \text{WW} \\ \text{tot} \end{cases} \quad (148)$$

folgt, daß die Frequenz des dynamischen, also des WW-Anteils, geringer ist als die kinematische Frequenz. Eine Beschränkung auf den dynamischen Anteil erlaubt also auch eine bessere zeitliche Auflösung. Man kann sogar über die Größe $\mathbf{V}\psi_S(t)$ die Stärke der Wechselwirkung bestimmen und daraus passende Zeitschritte berechnen, d. h. in Bereichen, in denen $\mathbf{V}\psi_S(t) \approx 0$ gilt, kann Δt sehr groß gewählt werden, in Bereichen mit starker Wechselwirkung dagegen ist ein kleinerer Wert für Δt sinnvoll.

Die Umrechnung vom Schrödingerbild in die WW-Darstellung erfolgt über

$$\psi_{\text{WW}}(t) = e^{iH_0 t} \psi_{\text{S}}(t) \quad (149)$$

$$= e^{iH_0 t} e^{-iHt} \psi(0), \quad (150)$$

mit H_0 als freiem Anteil des Hamiltonoperators $H = H_0 + V$.

Die zeitliche Entwicklung der Wellenfunktion im WW-Bild läßt sich bequem mit der SO-Methode beschreiben:

$$\psi_{\text{WW}}(t + \Delta t) \approx e^{-iH_0(t+\Delta t/2)} e^{-iV\Delta t} e^{-iH_0(t+\Delta t/2)} \psi_{\text{WW}}(t). \quad (151)$$

Details hierzu finden sich in [7].

4 Numerische Tests

4.1 Berechnung der Matrixexponentialfunktion

4.1.1 Allgemeines

Um verschiedene Verfahren hinsichtlich Speicherbedarf, Rechenaufwand und erzielbarer Genauigkeit vergleichen zu können, wurden sie in FORTRAN 77 implementiert. Die Berechnungen wurden mit den Datentypen COMPLEX und REAL und der Compileroption `-r8` durchgeführt, um doppelte Genauigkeit zu erhalten. Als Referenzergebnis diente eine parallel durchgeführte Berechnung der Matrixexponentialfunktion mit MATLAB 5 [36], die dort mit der in [37, Algorithmus 11.3-1] beschriebenen Methode des Skalierens und anschließender [6/6]-Padé-Approximation realisiert wurde.

Die Abweichung vom Referenzwert wurde mit der 1-Norm nach (42) berechnet, da diese Norm leicht in Programmen zu implementieren ist. Der relative Fehler (rF) berechnet sich gemäß

$$\text{rF} = \frac{\|e_{\text{approx}}^{-iA} - e^{-iA}\|}{\|e^{-iA}\|} \quad (152)$$

$$= \|e_{\text{approx}}^{-iA} - e^{-iA}\|, \quad (153)$$

da

$$\|e^{-iA}\| = 1 \quad (154)$$

gilt. Wird statt der 2-Norm $\|\cdot\|$ die 1-Norm $\|\cdot\|_1$ verwendet, so gilt (154) bis auf einen Faktor der Größenordnung $O(1)$, d. h. es gilt näherungsweise:

$$\text{rF} \approx \|e_{\text{approx}}^{-iA} - e_{\text{MATLAB}}^{-iA}\|_1. \quad (155)$$

Daraus läßt sich die Zahl der gültigen Ziffern (gZ) bestimmen, indem man

$$gZ = -\log_{10} \text{rF} \approx -\log_{10} \|e_{\text{approx}}^{-iA} - e_{\text{MATLAB}}^{-iA}\|_1 \quad (156)$$

setzt.

Als relative *Fehlerschranke* g wurde, wenn nicht anders im Text vermerkt, immer

$$g = 10^{-13}$$

gesetzt.

Als Testmatrizen wurden symmetrische (10×10) -Matrizen verwendet. Das gewünschte Spektrum wurde in die Hauptdiagonale geschrieben und die Matrix A anschließend mittels aufeinanderfolgender orthogonaler Transformationen durch Jacobi-Rotationen

eingesetzt. \bar{z} ist der Mittelpunkt des Spektrums von \mathbf{A} und Δz der Abstand vom Mittelpunkt zu den Grenzen (vgl. 2.1.3). Die Abbruchbedingung folgt aus den Entwicklungskoeffizienten nach Gl. (65), es gilt für $n > 0$:

$$|a_n| = 2 \cdot |J_n(\Delta z)| \quad (158)$$

Nachdem wegen der Skalierung $\|A'\| = 1$ und nach Definition $|T_n(x)| \leq 1$ für $|x| \leq 1$ ist, kann als Abbruchkriterium der doppelte Betrag der Besselfunktionen $J_n(\Delta z)$ dienen. Die Werte der Besselfunktionen wurden mit der Routine **S17DEF** aus der NAG-Bibliothek berechnet [38].

Im verwendeten Programm wurde das Abbruchkriterium dahingehend modifiziert, daß sowohl $2|J_N(\Delta z)| < g$ als auch $2|J_{N+1}(\Delta z)| < g$ erfüllt sein mußte, damit die Approximation bei N abgebrochen wurde. Dies liegt im Verhalten der Besselfunktionen als Funktion von n begründet (Abb. 2). Für $n < \Delta z$ zeigen diese oszillatorisches Verhalten, sodaß für ein bestimmtes n zufällig $J_n(\Delta z)$ nahe bei Null liegen könnte (siehe Abb. 2 bei $n = 8$). Erst für $n > \Delta z$ geht $J_n(\Delta z)$ exponentiell gegen Null. Obiges Abbruchkriterium hätte also auch als $2|J_N(\Delta z)| < g$ und zugleich $N > \Delta z$ formuliert werden können.

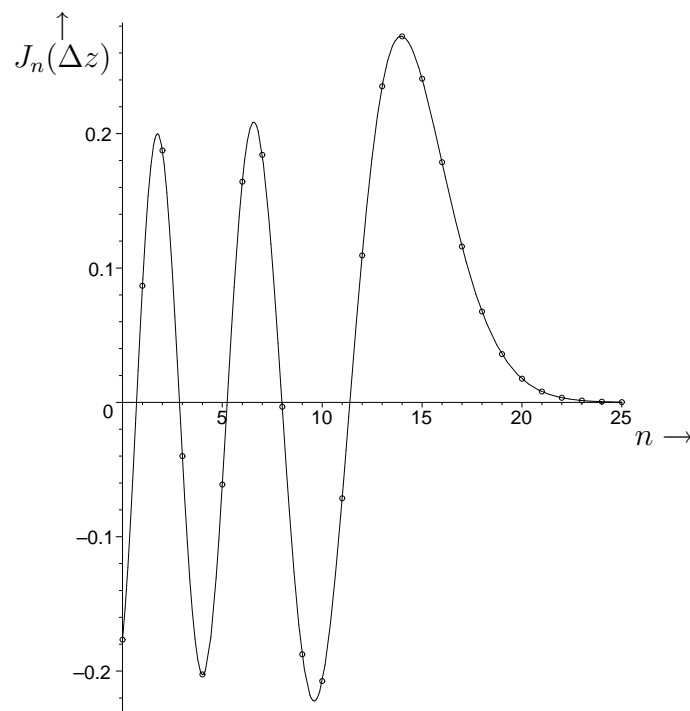


Abbildung 2: Verhalten der Besselfunktionen $J_n(\Delta z)$ als Funktion von n für $\Delta z = 16.02$

Die Ergebnisse der Berechnungen für die Testmatrizen zeigt Tab. 1.

Man sieht, wie die Anzahl der Approximationsschritte N von der Intervallbreite Δz abhängt; die erzielte Genauigkeit ist weitestgehend unabhängig vom Spektrum der

Matrix	Δz	N	gZ
A01	22.5	51	13.2
A02	22.5	51	13.0
A03	22.5	51	13.1
A04	22.5	51	12.7
A05	4.5	22	12.6

Tabelle 1: Ergebnisse für die Tschebyscheff-Approximation

Matrix. Lediglich die letzten beiden Matrizen weisen eine geringere Anzahl der gültigen Stellen auf. Gründe dafür sind zum einen die verwendete Näherung für gZ nach Gl. (156), zum anderen die unsymmetrische Eigenwertverteilung von A04, die durch die symmetrischen Tschebyscheff-Polynome nur in höherer Ordnung gut approximiert werden kann bzw. der schnelle Abfall der Besselkoeffizienten für A05, die ja skaliert dieselbe Struktur wie A01 aufweist.

Ein großer Nachteil des Tschebyscheff-Verfahrens ist die Tatsache, daß der Spektralbereich von A bekannt sein muß. Die Berechnung der Eigenwerte einer Matrix ist sehr rechenintensiv, eine Abschätzung u. U. fehlerhaft, besonders dann, wenn der Bereich etwas zu klein gewählt wurde. Ist Δz kleiner als die halbe Bereichsbreite, so folgt aus (157), daß

$$\|A'\| > 1 \quad (159)$$

ist. Für Argumente größer 1 wachsen die Tschebyscheffpolynome sehr schnell an, wie Abb. 3 für T_{10} zeigt (die Grenzen wurden nur um jeweils 1% verschoben!). Außerdem gehen die Entwicklungskoeffizienten für kleineres Δz schneller gegen Null, sodaß durch die geringere Anzahl der Approximationsschritte das Ergebnis ebenfalls verfälscht wird.

Um dennoch brauchbare Resultate zu erhalten, wurde versucht, das Abbruchkriterium zu erweitern, sodaß statt

$$(i) \quad 2|J_N(\Delta z)| < g \quad (160)$$

die Beziehung

$$(ii) \quad 2|J_N(\Delta z)| \cdot \|T_N(A')\|_1 < g \quad (161)$$

gelten muß, um beim Approximationsgrad N abzurechnen. Dies erhöht natürlich wieder den Rechenaufwand, da in jedem Approximationsschritt, der bereits Gl. (160) erfüllt, die Norm zu berechnen ist, um die Gültigkeit von (ii) zu überprüfen.

Im Vergleich dazu ist eine Überschätzung von Δz unkritisch, da dann $\|A'\| < 1$ gilt und kein „Überschießen“ der Tschebyscheff-Polynome an den Rändern die Ergebnisse beeinträchtigt. Es erhöht sich nur die Anzahl der Approximationsschritte, da für größere Δz der exponentielle Abfall dementsprechend später erfolgt.

Zur Überprüfung dieser Behauptungen wurden systematische Bereichsverschiebungen durchgeführt und verschiedene Abbruchbedingungen verwendet (als Testmatrix wurde A01 verwendet):

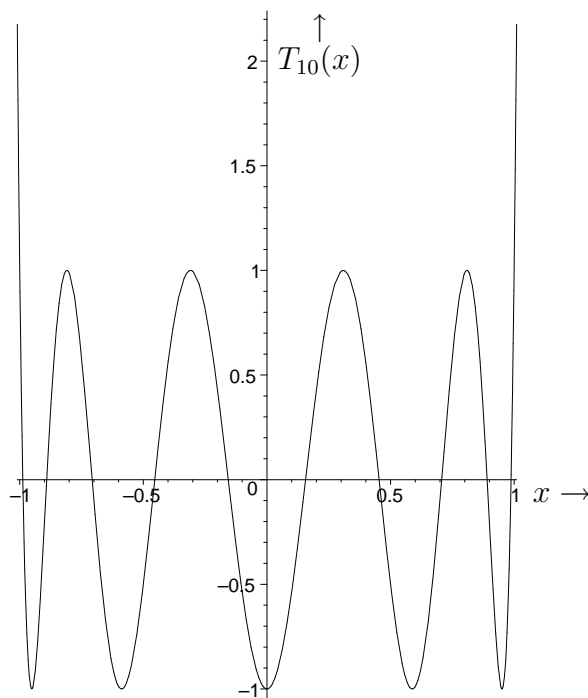


Abbildung 3: Tschebyscheffpolynom T_{10} im Intervall $[-1.01, 1.01]$

- (a) Symmetrische Verschiebung beider Grenzen, dadurch Bereichserweiterung um $x\%$; Abbruchkriterium (i)
- (b) Symmetrische Verschiebung wie bei (a); Abbruchkriterium (ii)
- (c) Unsymmetrische Bereichsverschiebung an der unteren Grenze; dies entspricht z. B. einer Vernachlässigung der Nullpunktenergie bei der Abschätzung (68) in 2.1.3); Erweiterung um $x\%$; Abbruchkriterium (i)
- (d) Unsymmetrische Verschiebung wie bei (c); Abbruchkriterium (ii)

Zu Vergleichszwecken (siehe 4.1.4) wurden für die skalierte Taylorentwicklung dieselben Bereichsverschiebungen durchgeführt:

- (e) Taylorentwicklung mit $q = 1$ und unsymmetrischer Bereichsverschiebung zum Vergleich

Abbildung 4 zeigt die Zahl der gültigen Stellen für die Fälle (a)-(e).

Man sieht, daß die Tschebyscheff-Approximation relativ unempfindlich gegenüber Bereichsüberschätzungen ist, wohingegen bei Unterschätzungen Fehler bis zu etwa 40% auftreten können. Außerdem sind die Fehler bei unsymmetrischer Verschiebung deutlich größer als bei symmetrischer Verschiebung. Die Genauigkeit läßt sich durch Verschärfung des Abbruchkriteriums durch Berücksichtigung der Norm um 8 bis 10 Größenordnungen verbessern (vgl. (a) mit (b) bzw. (c) mit (d) in Abb. 4). Dabei erhöht sich der Approximationsgrad jedoch um ca. 20% von $N = 51$ auf $N = 60$.

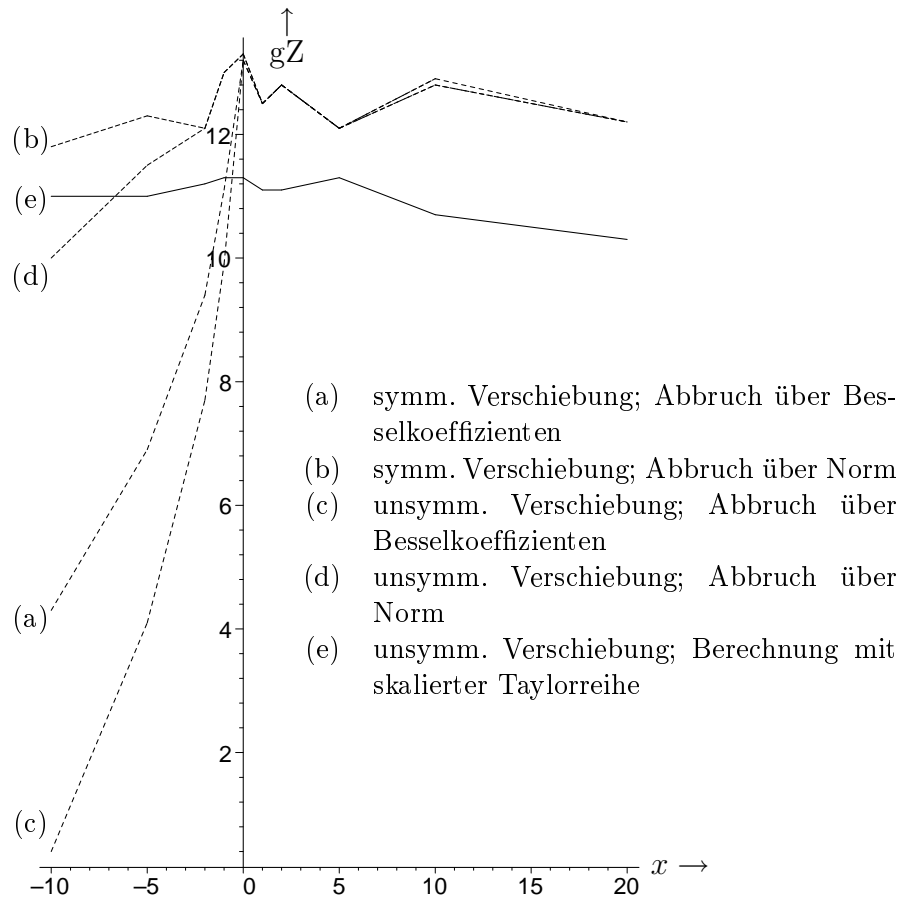


Abbildung 4: Zahl der gültigen Stellen gZ bei Bereichsverschiebungen um $x\%$ mit verschiedenen Abbruchkriterien. Bei unsymmetrischer Bereichsverschiebung wurde die untere Grenze verschoben; dadurch verlagert sich das Entwicklungszentrum \bar{z} nach oben.

4.1.3 Taylorapproximation

Die Approximation von $\exp(-iA)$ durch Taylorpolynome $p_n(-iA)$ wurde rekursiv durchgeführt, um Speicherplatz zu sparen, und zwar in der folgenden Weise:

$$\begin{aligned}
 p_{-1} &= 0, & t_0 &= 1 \\
 \left. \begin{aligned}
 p_n &= p_{n-1} + t_n \\
 t_{n+1} &= -\frac{iA}{n+1} t_n
 \end{aligned} \right\} & n &\geq 0.
 \end{aligned} \tag{162}$$

Als Abbruchkriterium wurde der Unterschied zweier aufeinanderfolgender Polynome herangezogen:

$$\|p_{n+1} - p_n\| = \|t_{n+1}\|. \tag{163}$$

Die Rekursion wurde also so lange durchlaufen, bis

$$\|t_{N+1}\|_1 < g \quad (164)$$

war. Dies entspricht einem Approximationsgrad N bei einer Genauigkeit g .

Hierbei traten Konvergenzprobleme auf, bedingt durch Rundungsfehler. Tabelle 2 zeigt die Ergebnisse.

Matrix	$\ A\ _1$	N	gZ
A01	45.9	133	-0.9
A02	62.3	133	-1.1
A03	48.7	133	-0.9
A04	55.0	132	-0.6
A05	23.9	82	6.9

Tabelle 2: Ergebnisse für die Taylorentwicklung um Null

Die Abweichungen vom Referenzwert sind beträchtlich, nur das Ergebnis für A05 hat immerhin ca. 7 gültige Stellen.

Dies ist auf die in 2.1.1 angesprochene langsame Konvergenz zurückzuführen. Für die Terme t_n gilt im skalaren Fall

$$|t_n| = \frac{x^n}{n!} \quad (165)$$

$$\approx \frac{x^n}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} = \left(\frac{x}{n}\right)^n \frac{e^n}{\sqrt{2\pi n}} \quad (166)$$

d.h. erst ab $n \approx x$ wächst die Fakultät im Nenner stärker als der Zähler. Für die Umformung (166) wurde die Stirlingformel

$$n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n} \left(1 + \frac{1}{12n} + \frac{1}{288n^2} + \dots\right) \quad (167)$$

verwendet.

Im vorliegenden Fall ist $x \approx \|A\|_1$, d.h., es sind für die Matrizen A01-A04 etwa 50-60 Terme zu berechnen, bevor Konvergenz eintritt. Dies ist in Abb. 5 für $x = 55$ illustriert. Wie man aus dieser Grafik entnimmt, steigen die Terme bis zu einem Betrag von $|t_n| \approx 10^{22}$ an, was in Verbindung mit dem alternierenden Vorzeichen zu großen Rundungsfehlern und damit verbundenen Stellenauslöschungen kommt, wodurch die Taylorreihe schließlich auf einen falschen Wert konvergiert.

Eine Verbesserung des Konvergenzverhaltens läßt sich erreichen, indem man die Exponentialfunktion nicht um Null herum entwickelt, sondern um den Mittelwert des Spektralbereiches. Ist z_{\min} , z_{\max} der kleinste bzw. größte Eigenwert, so folgt mit

$$\bar{z} = \frac{z_{\max} - z_{\min}}{2} \quad (168)$$

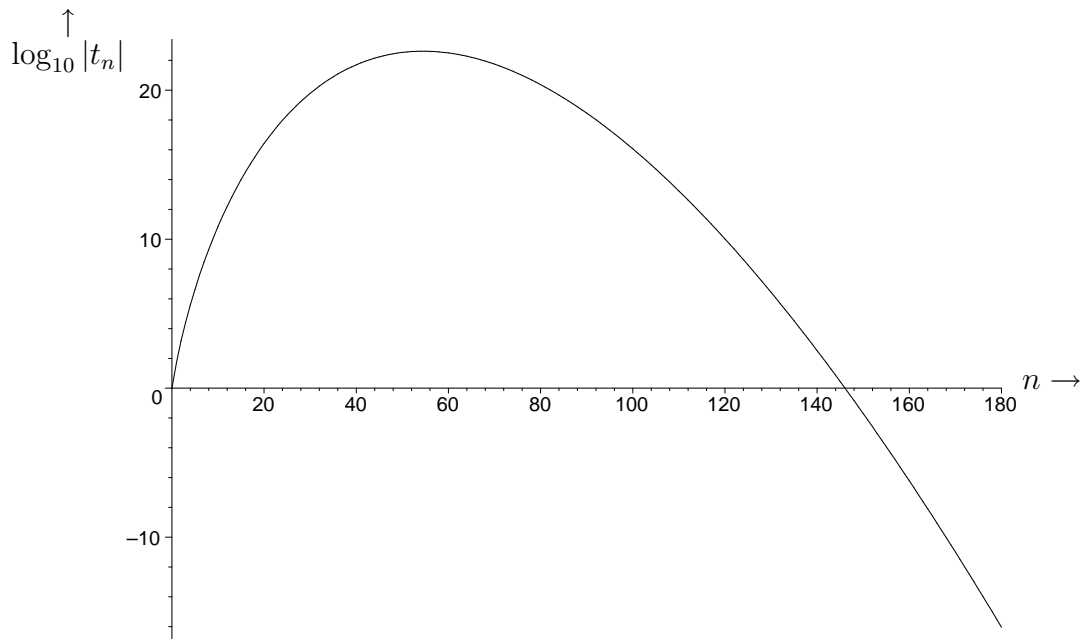


Abbildung 5: Logarithmus des Betrags der Terme $t_n = \frac{(-ix)^n}{n!}$ für $x = 55$. Das Maximum von $|t_n| \approx 10^{22}$ liegt in der Nähe von $n = x$. Erst für $n > x$ tritt Konvergenz ein.

und der Verschiebung

$$A \longrightarrow A' = A - \bar{z} \cdot 1 \quad (169)$$

die modifizierte Rekursion

$$\left. \begin{aligned} p'_{-1} &= 0, & t'_0 &= e^{-i\bar{z}} \cdot 1 \\ p'_n &= p'_{n-1} + t'_n \\ t'_{n+1} &= -\frac{iA'}{n+1} t'_n \end{aligned} \right\} n \geq 0 \quad (170)$$

führt.

Die damit erhaltenen Ergebnisse zeigt die Tabelle 3.

Wie man sieht, verkleinert sich die Norm der verschobenen Matrix um etwa \bar{z} , dadurch verbessert sich das Konvergenzverhalten.

Um die Konvergenzgeschwindigkeit zu erhöhen und damit Rundungsfehler weiter zu verringern, bietet sich die in 2.1.1 bereits angesprochene Möglichkeit des Skalierens mit $s = 2^q$ an. Bei den Berechnungen zu den nachfolgenden Ergebnissen wurde die Matrix A und der Mittelwert \bar{z} immer wieder halbiert und anschließend die Iteration (170)

Matrix	$\ A'\ _1$	N	gZ
A01	30.5	84	6.6
A02	45.3	84	6.6
A03	31.2	84	6.6
A04	37.5	84	6.3
A05	6.8	30	13.4

Tabelle 3: Taylorentwicklung um \bar{z}

durchgeführt, bis die Norm der reskalierten Matrix $< 1/2$ war. Um die Fehlernorm und die Anzahl der gültigen Ziffern zu bestimmen, wurden die Resultate anschliessend q mal quadriert. Die Ergebnisse im Vergleich mit den unskalierten Werten aus Tab. 3 zeigt Tabelle 4. Der Fehler läßt sich also bis zu einem gewissen Grad minimieren, steigt

q	A01		A02		A03		A04		A05	
	N	gZ	N	gZ	N	gZ	N	gZ	N	gZ
0	84	6.6	84	6.6	84	6.6	84	6.3	30	13.4
1	51	11.3	51	11.3	51	11.1	51	11.4	21	12.8
2	34	13.0	34	12.8	34	12.9	34	13.0	16	12.9
3	24	13.0	24	12.8	24	12.9	24	12.9	13	13.1
4	18	12.9	18	12.8	17	11.8	17	11.8	10	12.2
5	14	12.7	14	12.6	14	12.8	14	12.8	8	11.5
6	11	12.2	11	12.1	11	12.2	11	12.2	7	11.8
7	9	11.8	9	11.8	9	11.9	9	11.9	-	-

Tabelle 4: Ergebnisse bei Taylorapproximation mit Skalierungsfaktor $s = 2^q$

dann aber bei weiterer Skalierung wieder an und zeigt dabei oszillierendes Verhalten (im Ansatz sichtbar bei A05).

Um dies klarer zu zeigen, wurde für die Matrix A01 die Skalierung bis zu einem Skalierungsgrad von $q = 20$ durchgeführt. Das Ergebnis zeigt Abb. 6.

Der Grund für dieses Verhalten liegt darin begründet, daß durch die Verkleinerung der Norm der Matrix A das Abbruchkriterium nach weniger Approximationsschritten erfüllt ist, durch das nachfolgende Potenzieren aber geringe Abweichungen vom tatsächlichen Wert mitpotenziert und damit verstärkt werden. Abhilfe kann hier eine Verschärfung des Abbruchkriteriums bringen. Dazu ist aber anzumerken, daß die schnelle Konvergenz durch Skalierung durch einen erhöhten Rechenaufwand „erkauft“ wird und ein Iterationsschritt einer Quadrierung entspricht (jeweils eine Matrixmultiplikation). Dieser zusätzliche Aufwand muß natürlich bei Aussagen über die Effizienz mitberücksichtigt werden. Aufschluß hierüber gibt Abb. 7.

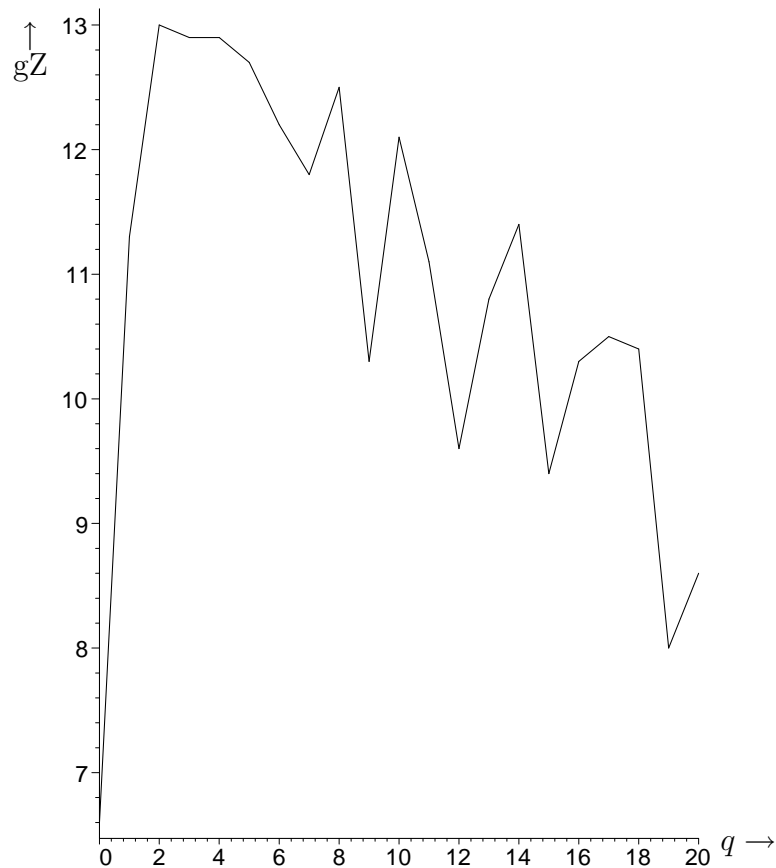


Abbildung 6: Anzahl der gültigen Ziffern gZ bei Taylorapproximation der skalierten Matrix $A/2^q$ in Abhängigkeit vom Skalierungsgrad q . Man erkennt für kleine q eine deutliche Zunahme der Genauigkeit; steigt q weiter an, so verringert sich die Anzahl der gültigen Stellen wieder. Dieser Abnahme ist außerdem ein oszillatorisches Verhalten überlagert.

4.1.4 Vergleich von Tschebyscheff- und Taylor-Approximation

Betrachtet man die Entwicklungskoeffizienten a_n der Tschebyscheffentwicklung, so sind diese (bis auf eine Konstante vom Betrag 1) gegeben durch $2 \cdot J_n(\Delta z)$. Die Besselfunktionen $J_n(x)$ lassen sich neben (40) auch in der folgenden Form darstellen:

$$J_n(x) = \frac{x^n}{2^n n!} \left(1 - \frac{x^2}{2(2n+2)} + \frac{x^4}{2 \cdot 4 \cdot (2n+2)(2n+4)} - \dots \right). \quad (171)$$

Für große n ($n \gg x$) ist dies

$$J_n(x) \approx \frac{x^n}{2^n n!} = \frac{\left(\frac{x}{2}\right)^n}{n!}. \quad (172)$$

Dies entspricht den Termen der Taylorreihe für $\exp(x/2)$. Zum Konvergenzverhalten der Tschebyscheff-Entwicklungskoeffizienten in x im Vergleich mit der Taylorreihe für

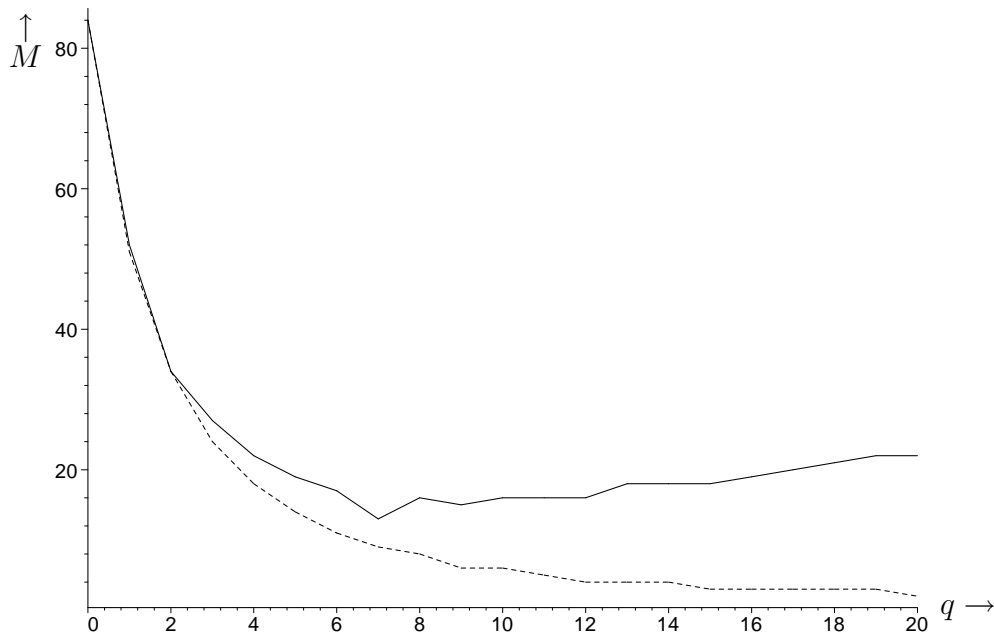


Abbildung 7: Durchgeführte Matrixmultiplikationen M in Abhängigkeit vom Skalierungsgrad q . Die punktierte Linie zeigt die Zahl der Multiplikationen, die notwendig ist, um Konvergenz zu erreichen. Die durchgezogene Linie berücksichtigt die notwendigen nachfolgenden Potenzierungen.

$x/2$ siehe Abb. 8.

Somit lassen sich Vergleiche bezüglich der Effizienz der beiden Approximationen bei vergleichbarem Rechenaufwand ziehen.

Anhand von Abb. 4 sieht man, daß die Taylorreihe weniger empfindlich auf Bereichsverschiebungen reagiert. Diese beeinflussen die Konvergenz nur im unsymmetrischen Fall, da nur dann das Entwicklungszentrum verschoben wird. Dies wirkt sich auf die Breite des Entwicklungsintervalls aus, sodaß indirekt die Konvergenzgeschwindigkeit verringert wird, was Rundungsfehler nach sich ziehen kann. So stieg die nötige Anzahl der Approximationsschritte von 51 auf 58 bei einer Verschiebung des Mittelpunkts der Matrix A01 um 20 %.

Bei vergleichbarem Rechenaufwand ist die Tschebyscheff-Approximation um etwa 1-2 Größenordnungen genauer als die Berechnung über die Taylorreihe (siehe Abb. 4 und Tab. 1 im Vergleich mit Tab. 4). Die Fehlernorm läßt sich aber durch weitere Skalierungen bis zu einem gewissen Grad noch minimieren, wobei gleichzeitig der Rechenaufwand sinkt (vgl. dazu Abb. 6 und 7). Damit lassen sich mit den Tschebyscheff-Ergebnissen vergleichbare Genauigkeiten erzielen. So sind beispielsweise bei der Matrix A01 mit der skalierten Taylorentwicklung ($q = 2$) $34+2=36$ Matrixmultiplikationen nö-

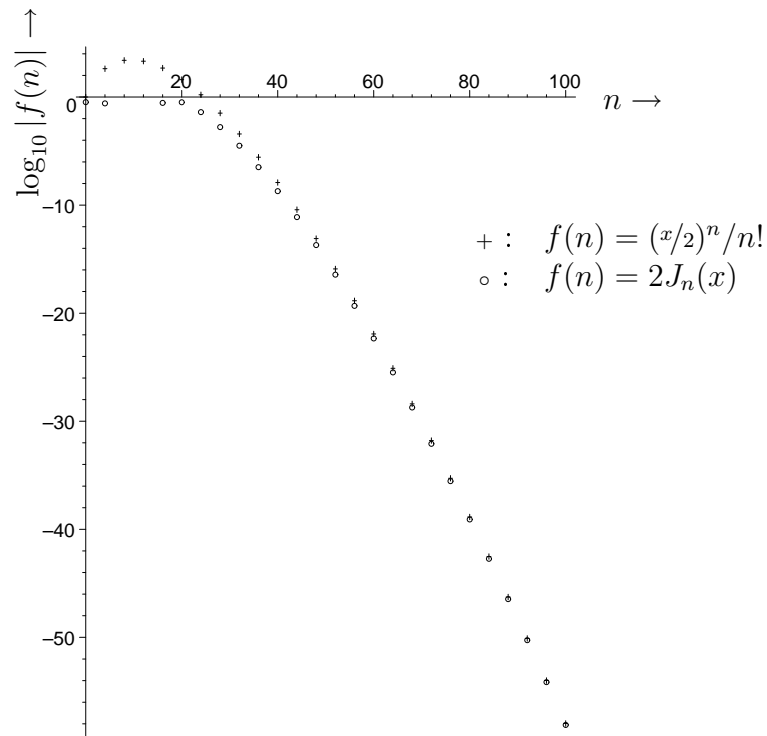


Abbildung 8: Konvergenzverhalten von $2J_n(x)$ und $(x/2)^n/n!$ in logarithmischem Maßstab für $x=20$.

tig (34 Matrixmultiplikationen $\rightarrow t_{35} < g$ und 2 Quadrierungen, siehe Tab. 4), um 13.0 gültige Stellen zu erhalten, während die Tschebyscheff-Entwicklung 51 Matrixmultiplikationen für 31.2 gültige Stellen benötigt (Tab. 1).

Ebenfalls zu berücksichtigen ist, daß die Rekursion der Taylorreihe so durchgeführt wurde, daß im n ten Schritt der $(n+1)$ -te Term berechnet wird, um die Anwendung von Beschleunigungsverfahren zu vereinfachen und besser vergleichen zu können (siehe 4.1.5; der Term t_{n+1} wird für die Vektor- \mathcal{J} -Transformation benötigt); dies bedeutet aber, daß als Abbruchkriterium der *nächste* noch nicht hinzuaddierte Term herangezogen wird, während im Falle der Tschebyscheff-Approximation der *letzte* bereits aufsummierte Term den Abbruch bestimmt. Dies beeinflusst auch die Genauigkeit der Berechnungen.

4.1.5 Beschleunigungsverfahren

Zunächst wurden Vorversuche mit MAPLE V Release 5.1 durchgeführt [39], um genauere Aussagen über die Effizienz der verschiedenen Beschleunigungsverfahren machen zu können. Dazu wurden die Taylormatrizen $p_n(-iA)$ in Vektoren umgewandelt und als Eingabedaten s_n an die Beschleunigungsroutinen übergeben. Um Vergleiche mit Ergebnissen aus der Literatur ziehen zu können und den Rechenaufwand über-

schaubar zu halten, wurde die Matrix

$$\mathbf{A} := \begin{pmatrix} -0.001 & -0.2 & -0.3 \\ -0.4 & -0.5 & -0.6 \\ -0.7 & -0.8 & -0.9 \end{pmatrix}$$

aus [15] verwendet. Diese Matrix besitzt die reellen Eigenwerte

$$\sigma(\mathbf{A}) \approx \{-1.60, 0.01, 0.19\}.$$

In der zitierten Arbeit wurde $\exp(\mathbf{A})$ mit ϵ -Algorithmen berechnet und im Vergleich zur reinen Taylorapproximation ein Stellengewinn von bis zu 5 Nachkommastellen erzielt. Ein ähnlicher Stellengewinn wurde auch für das vorliegende Problem erhofft, um den Rechenaufwand bei gleicher Genauigkeit gegenüber der reinen Taylorapproximation deutlich reduzieren zu können.

Als Beschleunigungsverfahren wurden zunächst

- der vektorielle ϵ -Algorithmus
- der erste topologische ϵ -Algorithmus
- und die vektorielle \mathcal{J} -Transformation mit der Hilfsfolge

$$r_n^{(k)} = \frac{1}{n + \beta_0 + \beta_1 \cdot k}$$

bzw. der daraus resultierenden Differenzenfolge

$$\Delta r_n^{(k)} = \frac{-1}{(n + \beta_0 + \beta_1 \cdot k)(n + 1 + \beta_0 + \beta_1 \cdot k)}$$

ausgewählt. Die verwendeten Algorithmen sind in Anhang A beschrieben. Die Testprogramme hatten die folgende Struktur:

1. Wahl des Entwicklungszentrums z
2. Wahl des (nächsten) Skalierungsfaktors und Skalierung von \mathbf{A} :

$$\mathbf{A} \longrightarrow \mathbf{A}' = \frac{\mathbf{A} - z\mathbf{1}}{2^q}$$

3. Berechnung der Startwerte:

$$p_{-1} = 0, \quad t_0 = e^{-iz}\mathbf{1}$$

4. Rekursionsschritt:

$$\begin{aligned} p_n &= p_{n-1} + t_n \\ t_{n+1} &= -\frac{i\mathbf{A}'}{n+1}t_n \end{aligned}$$

5. Umwandlung in Vektoren:

$$p_n \longrightarrow s_n$$

und ggf. (für die Vektor- \mathcal{J} -Transformation):

$$t_{n+1} \longrightarrow \omega_n$$

6. Aufruf der Beschleunigungsroutine

7. Umwandlung des resultierenden Vektors in eine quadratische Matrix

8. ggf. mehrfaches Quadrieren

9. Berechnung der Fehlernorm und der gültigen Ziffern

10. zurück zu Punkt 4, falls Abbruchkriterium noch nicht erfüllt ist

11. ggf. zurück zu Punkt 2 für die nächste Skalierungsstufe

Für den Vektor- \mathcal{J} -Algorithmus wurde zunächst $\beta_0 = 1$, $\beta_1 = 0$ gewählt.

Dabei zeigte sich, daß sich die Taylorreihe um den Mittelwert \bar{z} kaum beschleunigen läßt. Es wurden durchweg weniger als 2 Stellen Gewinn erzielt, meistens sogar weniger als 1 Stelle, bei Verwendung des ersten topologischen ϵ -Algorithmus war das Ergebnis sogar um bis zu 5 Stellen schlechter als die reine Taylorentwicklung. Auch die Vektor- \mathcal{J} -Transformation zeigte nur dann einen geringen Beschleunigungseffekt, wenn (wie oben im Ablaufplan der Testprogramme bereits angedeutet) als Restabschätzung ω_n der *nächste* Term t_{n+1} verwendet wurde, was in Verbindung mit der obengenannten Wahl der Konstanten β_0 und β_1 der \tilde{t} -Variante in Kap. 2.3.2, Gl. (104) entspricht⁹. Die Ergebnisse waren weitestgehend unabhängig davon, ob die Beschleunigung ab $n = 0$ oder erst ab einem bestimmten $n > 0$ durchgeführt oder nicht jedes Folgeelement, sondern jedes zweite, dritte usw. eingesetzt wurde.

Wurde das Entwicklungszentrum dagegen auf den oberen oder unteren Rand des Spektrums gelegt, war der Stellengewinn deutlich größer. Eine derartige Verschiebung hat zur Folge, daß alle Eigenwerte gleiches Vorzeichen besitzen und um Null herum entwickelt wird. Verstärkt wurde dieser Effekt durch ein Hochskalieren der Matrix A um einen Faktor > 1 , da dann die Taylorreihe von sich aus langsamer konvergiert und der Beschleunigungseffekt deutlicher zutage tritt. Mit der Vektor- \mathcal{J} -Transformation wurden bei der Entwicklung um den unteren Rand etwa 3 Stellen und bei Entwicklung um den oberen Rand 5 Stellen gewonnen. Beim ersten topologischen ϵ -Algorithmus

⁹Dieses Resultat wird verständlich, wenn man berücksichtigt, daß sich die Elemente einer konvergenzen Folge $\{s_n\}$ darstellen lassen als Summe aus dem Limes s und eines Restes R_n :

$$s_n = s + R_n.$$

Wählt man nun als Folgenglieder s_n die Partialsummen $s_n = \sum_{j=0}^n x^j/j!$, so gilt $s = e^x$ und $R_n = \sum_{j=n+1}^{\infty} x^j/j!$. Der betragsmäßig führende Term in der letzten Summe ist $x^{n+1}/(n+1)!$, was dem Term t_{n+1} entspricht.

waren am oberen Rand 1 bis 2 Stellen Zuwachs, am unteren Rand jedoch maximal 1 Stelle, im Verlauf der Rekursion dann eine Abnahme und sogar eine Verschlechterung gegenüber der reinen Taylorentwicklung zu beobachten.

Da der erste topologische ϵ -Algorithmus keine zufriedenstellenden Ergebnisse zeigte, wurde der zweite topologische ϵ -Algorithmus in die Reihe der Testprogramme mit aufgenommen. Der Algorithmus ist ebenfalls in Anhang A zu finden. Bei dieser Variante werden zur Berechnung im Gegensatz zum ersten topologischen ϵ -Algorithmus keine Elemente aus der vorletzten Gegendiagonale benötigt, sondern nur Elemente aus der letzten bzw. der gerade zu berechnenden. Dies entspricht etwa dem Unterschied zwischen der t - und der \tilde{t} -Variante (Kap. 2.3.2, Gl. (103) und (104)).

Der Unterschied im Verhalten an der oberen und der unteren Grenze von $\sigma(\mathbf{A})$ ist auf die Verteilung der Eigenwerte im Intervall $[z_{\min}, z_{\max}]$ zurückzuführen. Wird um die untere Grenze entwickelt, so entspricht dies einer Verschiebung der Matrix

$$\mathbf{A} \longrightarrow \mathbf{A}' = \mathbf{A} - z_{\min} \mathbf{1}$$

mit den Eigenwerten

$$\sigma(\mathbf{A}') \approx \{0, 1.61, 1.79\}, \quad (173)$$

während eine Entwicklung um die obere Grenze

$$\mathbf{A} \longrightarrow \mathbf{A}'' = \mathbf{A} - z_{\max} \mathbf{1}$$

und

$$\sigma(\mathbf{A}'') \approx \{-1.81, -1.79, 0\} \quad (174)$$

zur Folge hat. Im ersten Fall unterscheiden sich die beiden von Null verschiedenen Eigenwerte um etwa 10 %, im zweiten Fall dagegen nur um 1%. Daher ist zu vermuten, daß die Beschleunigungsverfahren besonders stark auf dominierende, d. h. weit außen liegende einzelne Eigenwerte oder auf eine Ansammlung von Eigenwerten in einem kleinen Intervallbereich ansprechen. Um diese Vermutung zu überprüfen, wurden die Algorithmen in FORTRAN 77 implementiert, da die Berechnungen zu aufwendig für eine Durchführung mit MAPLE wären.

Außerdem wurde versucht, den \mathcal{J} -Algorithmus dahingehend zu modifizieren, daß er weitgehend unabhängig von Lage und Vorzeichen der Eigenwerte beschleunigt. Hierzu wurde auf die allgemeine Definition in Kap. 2.4.3, Gl. (119) zurückgegriffen und als Matrixhilfsfolge

$$\mathbf{r}_n^{(k)} = r_n^{(k)} \mathbf{A} \quad \text{mit} \quad r_n^{(k)} = \frac{1}{n + 2 + (p - 1) \cdot k} \quad (175)$$

mit dem Parameter p gewählt.

Diese Modifikation beruht auf folgenden (vereinfacht dargestellten) Überlegungen: Grundlage aller (nichtskalaren) \mathcal{J} -Transformationen sind Modellfolgen der Form:

$$\mathbf{s}_n = \mathbf{s} + (c_0 + c_1 r_n) \omega_n \quad (176)$$

(vgl. [22, S. 210, Gl. (8.22)]). Dabei bezeichne \mathbf{s} den Limes der Matrizenfolge $\{\mathbf{s}_n\}$, ω_n eine geeignete Restabschätzung und \mathbf{r}_n eine Hilfsmatrixfolge. c_0 und c_1 sind (skalare) Konstanten. Ein Vergleich mit der Taylorreihe für die Matrixexponentialfunktion mit dem Argument \mathbf{A} liefert

$$\omega_n = t_{n+1} \quad \text{und} \quad c_0 = -1, \quad (177)$$

$$\mathbf{r}_n \omega_n = t_{n+2} \quad \text{und} \quad c_1 = -1. \quad (178)$$

Mit

$$t_{n+2} = \frac{\mathbf{A}}{n+2} t_{n+1} \quad (179)$$

folgt daraus

$$\mathbf{r}_n = \frac{\mathbf{A}}{n+2}, \quad (180)$$

was Gl. 175 mit $p = 1$ entspricht.

Der verwendete Algorithmus dieser *Matrix- \mathcal{J} -Transformation* ist in Anhang A zu finden. Hierbei wurden für $z = \bar{z}$ etwa 4 Stellen erzielt (mit $p = 1$). Zu beachten ist hierbei, daß hier die Größen p_n und t_n direkt als Matrizen übergeben werden, damit die notwendigen Matrixmultiplikationen durchgeführt werden können.

Für die Tests wurden die oben aufgeführten Beschleunigungsverfahren in FORTRAN 77 implementiert und bei verschiedenen Skalierungsstufen und Entwicklungszentren auf die Taylorreihen angewendet.

Dabei bestätigten sich im wesentlichen die bereits oben angedeuteten Ergebnisse. Abbildungen 9, 10 und 11 zeigen die Resultate für $q = 0, 3$ und 6 .

Es zeigt sich, daß der *erste topologische ϵ -Algorithmus* ($t\epsilon_1$) keine zufriedenstellenden Ergebnisse liefert; auch Versuche mit der Entwicklung um z_{\min} bzw. z_{\max} (auch in Verbindung mit anderen Eigenwertverteilungen wie bei A04) brachten keine Verbesserung des Konvergenzverhaltens. Somit ist dieser Algorithmus zur Beschleunigung der Exponentialfunktion ungeeignet.

Der *zweite topologische ϵ -Algorithmus* ($t\epsilon_2$) weist dagegen einen Beschleunigungseffekt auf. Der globale Verlauf ähnelt dem der unbeschleunigten Taylorentwicklung (gut sichtbar bei Abb. 9), dabei treten jedoch Oszillationen auf, die für höhere Skalierungsstufen dazu führen, daß der $t\epsilon_2$ um die Taylorkurve pendelt (siehe Abb. 10) und somit u. U. schlechtere Ergebnisse als die Ausgangsreihe liefert.

Als relativ unempfindlich gegenüber der Wahl des Entwicklungszentrums und der Verteilung der Eigenwerte stellte sich der *vektorielle ϵ -Algorithmus* ($v\epsilon$) dar; typisch sind die Oszillationen in dem Bereich, in dem die Taylorreihe anwächst und das Einsetzen der Beschleunigung ab dem Punkt, an dem auch die Taylorreihe zu konvergieren beginnt. Dieses Verhalten legt es nahe, den $v\epsilon$ erst ab $N \approx \|A\|$ einsetzen zu lassen, um den Rechenaufwand zu verringern. Der $v\epsilon$ beschleunigt dann noch stärker auf das Endergebnis hin. Alle anderen getesteten Beschleunigungsverfahren zeigten dabei nur einen geringen Effekt, in einigen Fällen erhöhte sich die Approximationsordnung sogar. Der Gewinn durch $v\epsilon$ entspricht etwa der Verringerung der Approximationsordnung durch das Skalieren um den Faktor 2.

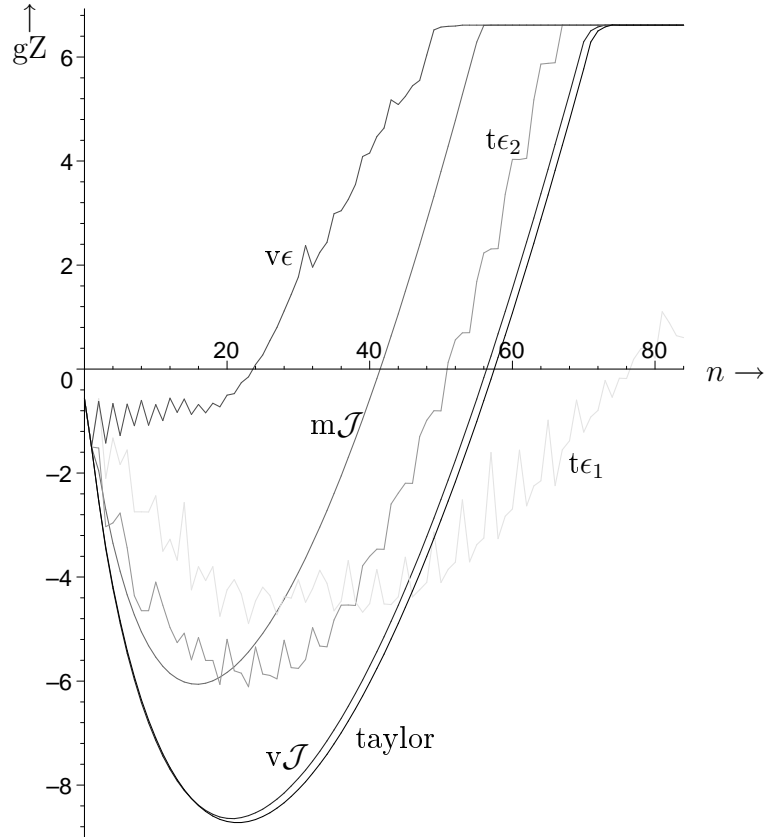


Abbildung 9: Ergebnisse der Beschleunigungsverfahren bei Matrix A01 um \bar{z} mit $q = 0$. Die Abszisse zeigt die Anzahl der durchgeführten Approximationsschritte n , die Ordinate die dabei erhaltene Anzahl der gültigen Stellen gZ .

Auffällig für den *vektoriellen \mathcal{J} -Algorithmus* ($v\mathcal{J}$) ist sein glatter Verlauf, der anfangs fast deckungsgleich über der Taylorreihe liegt und kurz vor dem Einsetzen der Konvergenz abweicht, um dann parallel dazu weiterzulaufen (siehe Abb. 9). Für höhere Skalierungsstufen liefert der $v\mathcal{J}$ ähnlich gute Ergebnisse wie der $v\epsilon$. Liegt ein isolierter Eigenwert vor, so ändert sich das Verhalten des $v\mathcal{J}$ signifikant; dabei ist das Vorzeichen des Eigenwertes ohne Bedeutung. Dies sei an Abbildung 12 im Vergleich zum vektoriellen ϵ -Algorithmus illustriert. Links ist die Matrix A04, die einen isolierten Eigenwert besitzt, mit $q = 1$ skaliert und um \bar{z} entwickelt. Das typische Verhalten von $v\epsilon$ bzw. $v\mathcal{J}$ ist deutlich zu erkennen. Rechts dagegen wurde um z_{\max} entwickelt (die Skalierung wurde auf $q = 2$ gesetzt, um beide Kurvenverläufe direkt vergleichen zu können). Während der Verlauf des $v\epsilon$ im wesentlichen unverändert bleibt (nur die Oszillationen zu Beginn fallen etwas stärker aus), beginnt $v\mathcal{J}$ bereits bei $n = 8$ zu konvergieren und liegt deutlich vor $v\epsilon$.

Der *Matrix- \mathcal{J} -Algorithmus* ($m\mathcal{J}$) weist einen dem $v\mathcal{J}$ vergleichbaren Kurvenverlauf bei deutlich stärkerer Beschleunigung auf (hier mit $p=2$); durch die zusätzlich anfallenden Matrixmultiplikationen ($2(n + 1)$ beim n . Aufruf der Routine) ist dieses

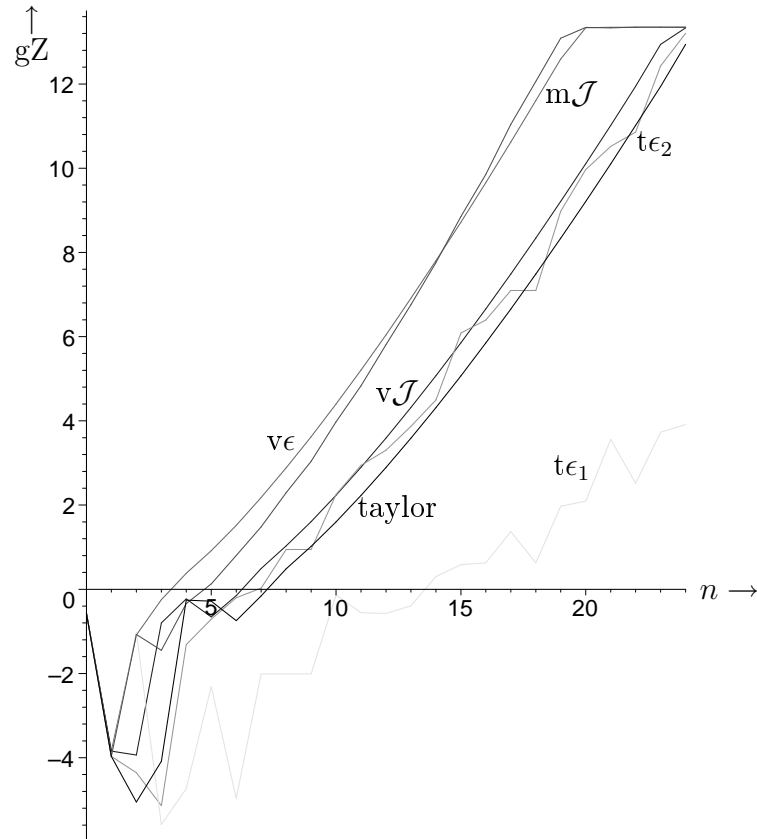


Abbildung 10: Ergebnisse der Beschleunigungsverfahren bei Matrix A01 um \bar{z} mit $q = 3$. Abszisse: Zahl der Approximationsschritte n , Ordinate: gültige Ziffern gZ .

Verfahren jedoch deutlich rechenintensiver als die oben genannten. Bei höheren Skalierungen reduziert sich außerdem der Beschleunigungseffekt, sodaß dieses Verfahren keine grundsätzlichen Vorteile mit sich bringt.

Beide \mathcal{J} -Algorithmen wurden auch mit unterschiedlichen Parametern $\beta_0 = 1, \beta_1 = 0, 1, 2$ bzw. $p = 1, 2, 3$ getestet. Die Unterschiede liegen aber im Bereich $0.01 gZ$ und sind deshalb vernachlässigbar.

Für alle oben genannten Beschleunigungsverfahren wurde außerdem getestet, wie sich das Beschleunigungsverhalten ändert, wenn nicht für jedes n die jeweilige Beschleunigungsroutine aufgerufen wird, sondern nur für jedes zweite, dritte, usw. Für den Aufruf bei jedem zweiten n sind die Ergebnisse bei allen Routinen ähnlich, somit läßt sich der Rechenaufwand bei gleichbleibender erzielbarer Genauigkeit weiter minimieren; bei höherer Schrittweite verschlechterte sich das Konvergenzverhalten zunehmend.

Die drei letztgenannten Beschleunigungsverfahren lieferten bei höheren Skalierungen ein genaueres Ergebnis als die reine Taylorentwicklung (ca. 1.5 Stellen mehr bei $q = 7$, siehe auch Abb. 11). Auch $t\epsilon_2$ zeigte dieses Verhalten, das durch die oben erwähnten Oszillationen allerdings teilweise überdeckt wurde.

Zusammenfassend läßt sich sagen, daß sich Beschleunigungsverfahren gut zur Berech-

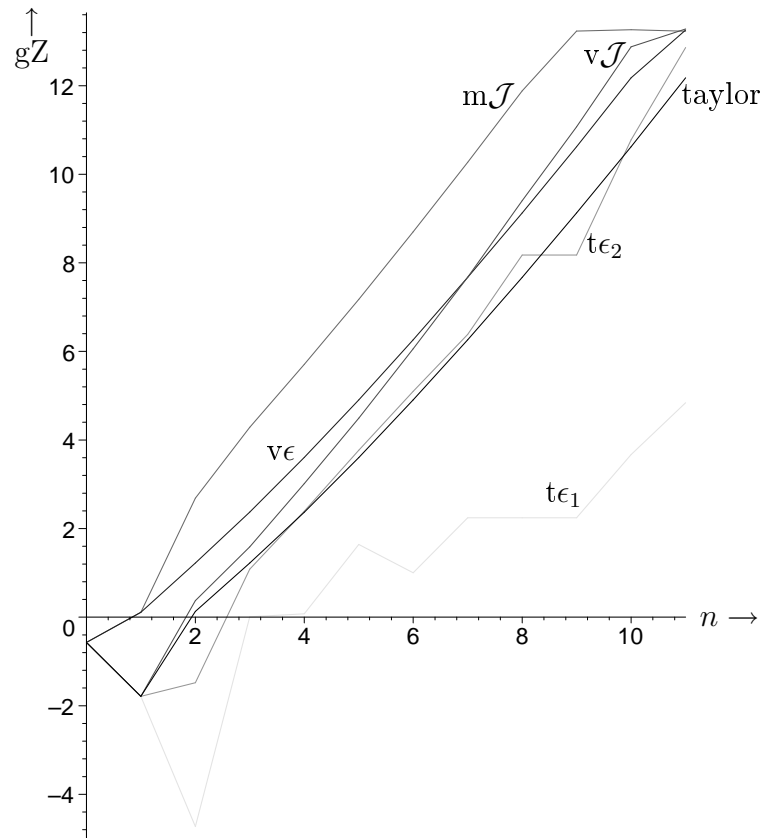


Abbildung 11: Ergebnisse der Beschleunigungsverfahren bei Matrix A01 um \bar{z} mit $q = 6$. Abszisse: Zahl der Approximationsschritte n , Ordinate: gültige Ziffern gZ .

nung der Matrixexponentialfunktion einsetzen lassen. Die Vorversuche und Tests haben ergeben, daß der vektorielle ϵ -Algorithmus und der vektorielle \mathcal{J} -Algorithmus die geeignetsten Routinen zur Beschleunigung darstellen. Dabei liefert das ϵ -Verfahren unabhängig von der Struktur der Matrix gute Ergebnisse, während die \mathcal{J} -Methode ihre Stärken besonders bei isolierten Eigenwerten zeigt.

Dabei sollte allerdings nicht vergessen werden, daß bei der Verwendung von Beschleunigungsverfahren grundsätzlich der Speicherbedarf größer ist als beim Tschebyscheff-Verfahren, da zumindest die letzte Gegendiagonale in der ϵ - bzw. der \mathcal{J} -Tafel abgespeichert werden muß (siehe dazu auch Anhang A).

Konkret kann die Berechnung der Matrixexponentialfunktion dergestalt durchgeführt werden, daß die Matrix bis $\|A\|_1 < 1/2$ herunterskaliert wird und der sich ergebende Verlust an Genauigkeit (siehe dazu Abb. 6) durch ein Beschleunigungsverfahren kompensiert wird. Bei so hohen Skalierungsstufen (im vorliegenden Fall $q = 7$) kann als Entwicklungszentrum neben dem Mittelwert \bar{z} auch die Spur $\text{tr}(A)$, geteilt durch die Dimension D der Matrix, verwendet werden. Somit ist eine Berechnung mit einer Genauigkeit möglich, die mit der der Tschebyscheff-Approximation vergleichbar ist, ohne die Struktur bzw. das Spektrum der Matrix zu kennen.

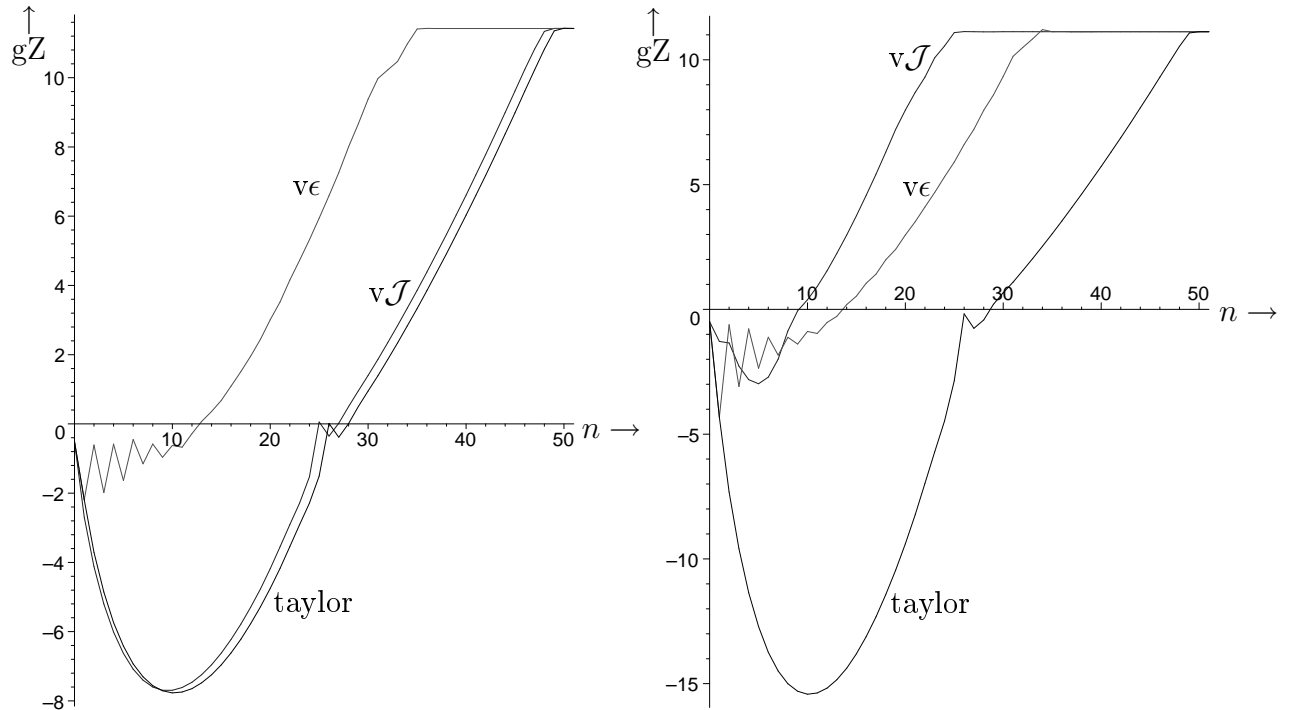


Abbildung 12: Vergleich von vektoriellem ϵ - und \mathcal{J} -Algorithmus in Abhängigkeit von der Eigenwertverteilung. Aufgetragen ist jeweils die Anzahl der gültigen Ziffern (gZ) gegen die Approximationsordnung n . Links wurde mit $q = 1$ skaliert und um \bar{z} entwickelt, rechts dagegen mit $q = 2$ um z_{\max} . Man beachte das unterschiedliche Verhalten der vektorielle \mathcal{J} -Transformation ($v\mathcal{J}$).

4.2 Vektorpropagation

4.2.1 Vorbemerkungen

Auch hier wurden die Programme in FORTRAN 77 implementiert und als Referenzdaten die MATLAB-Ergebnisse verwendet. Als Testvektor wurde

$$\psi(0) = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (181)$$

herangezogen. Die Berechnungen wurden mit den bereits aus Abschnitt 4.1 bekannten Matrizen A01-A05 durchgeführt.

4.2.2 Tschebyscheff-Approximation

Hier wurden die Tschebyscheff-Matrizen durch Vektoren ersetzt und die Rekursion (137) aus 3.1.3 verwendet:

$$\begin{aligned}\psi_0 &= \psi(0), & \psi_1 &= \mathbf{H}_{\text{norm}}\psi(0) \\ \psi_{n+1} &= 2\mathbf{H}_{\text{norm}}\psi_n - \psi_{n-1} & (n \geq 1).\end{aligned}$$

Die Ergebnisse für $\psi(0)$ nach (181) sind vergleichbar mit denen aus 4.1.2 hinsichtlich Genauigkeit, Rechenaufwand und Fehlertoleranz. Statt der Matrixnorm $\|T_N(A')\|_1$ wurde die Vektornorm $\|\psi_N\|_1$ verwendet (zur Definition siehe (42)).

4.2.3 Beschleunigungsverfahren

Grundlage für die Anwendung von Beschleunigungsverfahren ist wieder die Taylorreihe. Die Rekursion (162) lautet, umgeschrieben auf Vektoren:

$$\begin{aligned}\psi_{-1} &= 0, & \tau_0 &= e^{-iz}\psi(0) \\ \left. \begin{aligned}\psi_n &= \psi_{n-1} + \tau_n \\ \tau_{n+1} &= -\frac{iA'}{n+1}\tau_n\end{aligned} \right\} & n &\geq 0.\end{aligned}\tag{182}$$

Genauso wie im Matrizenfall läßt sich eine schnellere Konvergenz durch geeignetes Skalieren der Matrix \mathbf{A} erreichen; hier bedeutet jedoch eine Skalierung um einen Faktor s , daß die Propagation s mal hintereinander durchgeführt werden muß. Setzt man $s = 2^q$ und vergleicht mit den Daten aus Tabelle 4, so ergibt sich unskaliert ein Approximationsgrad von $N = 84$, für $q = 1$ schon $N = 2 \cdot 51 = 102$ usw. bis hin zu $N = 2^7 \cdot 9 = 1152$ für $q = 7$. Man sieht also, daß der Rechenaufwand durch mehrfache Propagation trotz schnellerer Konvergenz sehr stark ansteigt. Anders als im Matrizenfall kann hier also nicht beliebig herunterskaliert werden. Zu beachten ist allerdings, daß bei der Vektorpropagation die Wahl $s = 2^q$ nicht zwingend ist, so lange s nur positiv und ganzzahlig ist. Deshalb wurden im folgenden auch die Ergebnisse zu $s = 3$ mit aufgenommen; dies entspricht einer Potenz q von

$$q = \frac{\log 3}{\log 2} \approx 1.585.$$

Um den Rechenaufwand zu reduzieren, bietet sich die Verwendung von Beschleunigungsverfahren an. Wie bereits in 4.1.5 diskutiert, sind der vektorielle ϵ - bzw. \mathcal{J} -Algorithmus geeignete Verfahren. Als Abbruchkriterium wurde die Ungleichung

$$\|\psi_N - \psi_{N-1}\|_1 < g \tag{183}$$

mit den Vektoren ψ_N aus der Taylorrekursion bzw. den von den Beschleunigungsroutinen zurückgelieferten Vektoren verwendet.

q	$s = 2^q$	Taylor		$v\epsilon$		$v\mathcal{J}$	
		N	gZ	N	gZ	N	gZ
0	1	83	7.2	65	7.2	81	7.2
1	2	100	11.9	70	12.2	98	11.8
1.585	3	117	12.9	84	13.3	114	13.3
2	4	132	12.8	100	13.8	128	13.8
3	8	184	12.5	152	13.9	176	13.5
4	16	272	12.4	240	14.0	256	13.3
5	32	416	12.0	384	13.7	384	13.0
6	64	640	11.2	640	13.7	640	13.6
7	128	1152	12.4	1152	14.0	1024	13.4

Tabelle 5: Tabelle zum Vergleich von Rechenaufwand und erzielter Genauigkeit von Taylorapproximation, vektoriellem ϵ -Algorithmus ($v\epsilon$) und vektorieller \mathcal{J} -Transformation ($v\mathcal{J}$) bei der Vektorpropagation und verschiedenen Skalierungsgraden $s = 2^q$: Matrix A01 um \bar{z} entwickelt

Einen Vergleich für die Matrix A01 und den Vektor $\psi(0)$ nach (181) liefert Tabelle 5. Man sieht, daß für kleine q die Beschleunigungsroutinen den Rechenaufwand verringern, für größere q dagegen der Rechenaufwand (fast) gleich bleibt, dafür aber die Genauigkeit (ggü. der Taylorapproximation) anwächst. Im vorliegenden Beispiel wäre eine Skalierung mit $q = 3$ und die Verwendung des vektoriellen ϵ -Algorithmus am günstigsten, da bei 152 Approximationsschritten 13.9 gültige Ziffern erreicht werden; eine Verbesserung um 0.1 gZ (entspricht 0.7%) erfordert einen um etwa 58% größeren Rechenaufwand.

Sind die Eigenwerte nicht gleichmäßig verteilt, so ergeben sich andere Resultate, wie in den Tabellen 6 und 7. Aus Tabelle 6 ist ersichtlich, daß auch hier bei bekannter

q	$s = 2^q$	Taylor		$v\epsilon$		$v\mathcal{J}$	
		N	gZ	N	gZ	N	gZ
0	1	83	7.7	66	7.7	80	7.7
1	2	100	12.0	72	11.8	96	12.0
1.585	3	177	12.9	84	13.2	111	13.4
2	4	132	12.8	100	13.5	124	13.5
3	8	184	12.6	152	13.6	168	13.2
4	16	272	12.4	240	13.6	256	13.8
5	32	416	12.1	384	13.5	384	13.5
6	64	640	11.3	640	13.5	621	13.2
7	128	1152	12.5	1152	13.7	1024	13.8

Tabelle 6: Tabelle zum Vergleich von Rechenaufwand und erzielter Genauigkeit: Matrix A04 um \bar{z} entwickelt

q	$s = 2^q$	Taylor		$v\epsilon$		$v\mathcal{J}$	
		N	gZ	N	gZ	N	gZ
0	1	119	2.8	108	2.8	88	2.8
1	2	136	10.8	102	10.7	79	11.1
1.585	3	153	13.0	102	13.2	81	13.1
2	4	168	13.0	119	13.4	88	13.5
3	8	216	12.3	168	13.1	120	13.2
4	16	304	12.2	256	13.2	192	13.7
5	32	448	12.0	416	13.6	320	13.5
6	64	704	12.0	640	12.9	576	13.8
7	128	1152	12.1	1152	13.7	1024	13.8

Tabelle 7: Tabelle zum Vergleich von Rechenaufwand und erzielter Genauigkeit: Matrix A04 um $\text{tr}(\mathbf{A})/D$ entwickelt

Eigenwertverteilung der vektorielle ϵ -Algorithmus die besten Ergebnisse liefert, obwohl hier eindeutige Aussagen schon schwerer zu treffen sind (vgl. $q = 4$: $N = 240$ bei $v\epsilon$ und 13.6 gZ ggü. $N = 256$ und 13.8 gZ bei $v\mathcal{J}$).

Entwickelt man dagegen um $\text{tr}(\mathbf{A})/D$ (D : Matrixdimension)¹⁰, so sieht man anhand von Tabelle 7, daß die vektorielle \mathcal{J} -Transformation deutlich besser abschneidet und sogar den ϵ -Algorithmus bei der Entwicklung um \bar{z} überholt ($q = 2$: jeweils 13.5 gZ; Rechenaufwand $v\epsilon$: $N = 100$ ggü. $N = 88$ bei $v\mathcal{J}$).

4.2.4 Vergleich von Tschebyscheff- und Beschleunigungsverfahren

Vergleicht man nun die Effizienz der verschiedenen Vektorpropagationsmethoden, so läßt sich auch durch den Einsatz von Beschleunigungsalgorithmen der Rechenaufwand nicht so weit verringern, daß er mit der Tschebyscheff-Entwicklung vergleichbar wäre. Der Grund dafür liegt in der langsamen Konvergenz der Taylorreihe, deren Terme einen weiten Stellenbereich überstreichen, sodaß aufgrund der endlichen Stellenzahl im Computer Skalierungen notwendig werden, die die Zahl der Approximationsschritte in die Höhe treiben.

Anhand der Tabellen 5-7 wird auch ersichtlich, daß der Einsatz der Vektorpropagation kein „black box“-Verfahren darstellt, das wie im Matrixfall (siehe Anmerkungen am Ende von Abschnitt 4.2.3) unabhängig von der Problemstellung angewandt werden kann; vielmehr ist die Wahl des Skalierungsgrads s und (je nach Struktur der Matrix) des verwendeten Algorithmus sehr entscheidend und muß im Einzelfall eventuell durch geeignete Vorversuche überprüft werden.

Es zeigt sich im folgenden (siehe Abschnitt 5.1.3), daß in der Praxis die Beschleunigungsverfahren durchaus eine ernstzunehmende Alternative zum Tschebyscheff-Ver-

¹⁰Dies entspricht dem arithmetischen Mittel der Eigenwerte.

fahren darstellen können, wenn nicht nur die Rechenzeit und der Speicherbedarf, sondern auch die erzielte Genauigkeit berücksichtigt wird.

5 Anwendungen auf Modellsysteme

5.1 Die Zeitentwicklung von I_2

5.1.1 Motivation und physikalische Grundlagen

Als einfaches physikalisches Problem wurde ein zweiatomiges Molekül gewählt, das als einzigen inneren Freiheitsgrad den Abstand r der Atome voneinander besitzt. Konkret wurden die Berechnungen an einem I_2 -Molekül durchgeführt. Dieses Molekül wird im übernächsten Abschnitt (Kap. 5.3) zu dem van der Waals-Molekül HeI_2 erweitert, sodaß die in diesem Abschnitt vorgestellten Rechnungen als Vorversuche zu sehen sind, um an einem einfachen Problem mit den bisher theoretisch behandelten Methoden vertraut zu werden.

Der Hamiltonoperator dieses Systems ist durch

$$\hat{H} = -\frac{1}{2m} \frac{\partial^2}{\partial r^2} + \hat{V}_{I-I}(r) \quad (184)$$

gegeben. Die potentielle Energie wird üblicherweise¹¹ durch ein Morse-Potential der Form

$$V_{I-I}(r) = D_r \left[e^{-2\alpha_r(r-r_0)} - 2e^{-\alpha_r(r-r_0)} \right] \quad (185)$$

beschrieben. Im folgenden wird statt $V_{I-I}(r)$ der Einfachheit halber immer $V(r)$ ohne Indizierung gesetzt. Die Energieeigenwerte für einen Hamiltonoperator der Form (184) sind analytisch darstellbar durch

$$E_v = \omega_0 \left(v + \frac{1}{2} \right) - \omega_0 x \left(v + \frac{1}{2} \right)^2 - D_r, \quad (186)$$

wobei v die Anzahl der Schwingungsquanten angibt. Die Bestimmungsgrößen ω_0 und x berechnen sich gemäß

$$\omega_0 = \sqrt{\frac{2D_r\alpha_r^2}{m}} \quad (187)$$

$$x = \frac{\omega_0}{4D_r}. \quad (188)$$

Auch die Eigenfunktionen ψ_v zu den Eigenwerten E_v sind analytisch darstellbar. Für ein Morsepotential mit den Parametern D_r , α_r und der Masse m bestimmt die Konstante

$$\kappa = 2 \frac{\sqrt{2mD_r}}{\alpha_r \hbar} \quad (189)$$

¹¹In [40] wird darauf hingewiesen, daß eine Beschreibung des Potentials in der Form (185) physikalisch nicht korrekt ist, da ein endliches Potential an der Stelle $r = 0$ einen von Null verschiedenen Wert der Wellenfunktion an dieser Stelle impliziert und somit die Randbedingung bei $r = 0$ verletzt ist. In der Praxis sind die Werte in der Nähe von $r = 0$ jedoch genügend groß, sodaß die Wellenfunktion an dieser Stelle in guter Näherung verschwindet. Der große Vorteil der Verwendung eines Morsepotentials ist die analytische Darstellung der Energieeigenwerte und der Wellenfunktionen, was die mathematische Behandlung des Problems stark vereinfacht.

die Anzahl der möglichen gebundenen Zustände [40]. Es gilt der Zusammenhang

$$0 \leq v \leq \frac{\kappa - 1}{2}, \quad (190)$$

wobei die Schwingungsquantenzahl v ganzzahlig sein muß.

Definiert man die verallgemeinerten Laguerre-Polynome gemäß¹²

$$L_{\kappa-v-1}^{\kappa-2v-1}(x) = \sum_{j=0}^v \frac{\Gamma(v+j-\kappa+1)}{(v-j)!j!} x^{v-j}, \quad (191)$$

so ist die vollständige Wellenfunktion in der Koordinate r (bis auf einen Normierungsfaktor) gegeben durch

$$|v\rangle = \psi_v(r) \propto e^{-\frac{\kappa}{2}e^{-\alpha(r-r_0)}} \cdot \left[\kappa e^{-\alpha(r-r_0)} \right]^{(\kappa-2v-1)/2} \cdot L_{\kappa-v-1}^{\kappa-2v-1} \left(\kappa e^{-\alpha(r-r_0)} \right). \quad (192)$$

Abbildung 13 zeigt die typische Form des Morsepotentials, die Energieeigenwerte und die Eigenfunktionen (nicht normiert). Die verwendeten Parameter sind unter der Grafik angegeben.

Für Jod betragen die Werte der Parameter nach [41] (ebenso zu finden in [42, 43, 44, 45, 46]):

$$m = 63.5 \text{ amu} \quad (193)$$

$$D_r = 4911 \text{ cm}^{-1} \quad (194)$$

$$\alpha_r = 0.9380 \text{ a}_0^{-1} \quad (195)$$

$$r_0 = 5.6994 \text{ a}_0. \quad (196)$$

Daraus ergeben sich¹³

$$\omega_0 = 128 \text{ cm}^{-1} \quad (197)$$

$$x = 6.52 \cdot 10^{-3} \quad (198)$$

$$\kappa = 153.5 \Rightarrow 0 \leq v_r \leq 76. \quad (199)$$

Als Ausgangswellenfunktion wurde eine Überlagerung der Jod-Eigenzustände gemäß

$$\psi_0 = \frac{1}{\sqrt{2}} (|5\rangle + |6\rangle) \quad (200)$$

gewählt.

¹²In [40] wird neben einem zusätzlichen Normierungsfaktor eine komplexe Phase in die Definition eingefügt, die allerdings o. B. d. A. gleich Eins gesetzt werden kann, um die Wellenfunktion reell zu machen.

¹³ ω_0 ist hier in Einheiten von cm^{-1} angegeben, da das Potential üblicherweise in dieser Einheit vorliegt. 1 eV entspricht 8065.5 cm^{-1} .

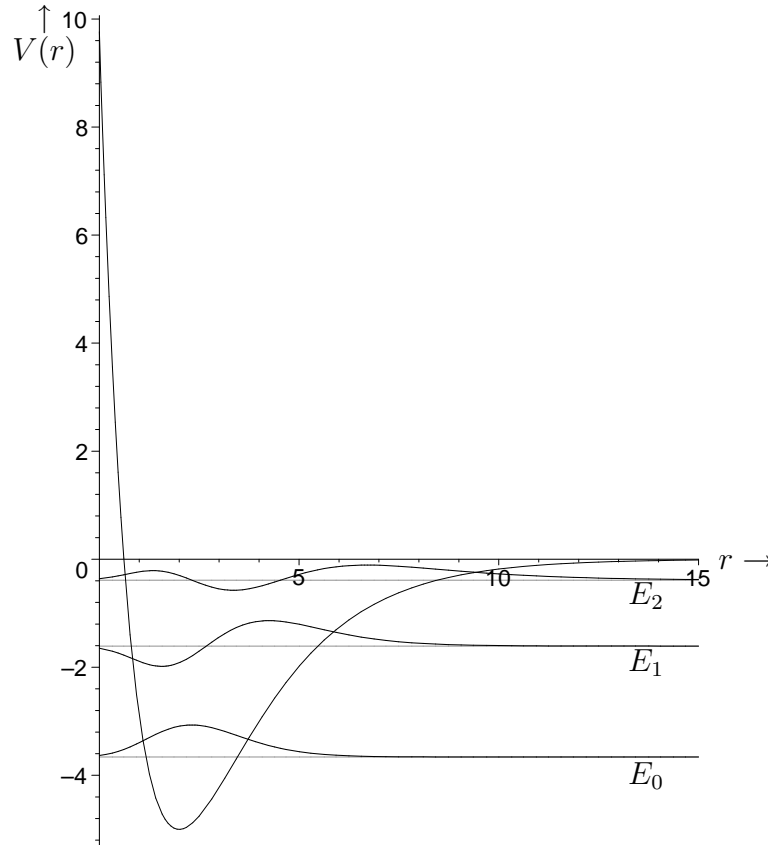


Abbildung 13: Potentialform, Energieeigenwerte und Eigenfunktionen eines Morsepotentials mit $D_r = 5 \text{ cm}^{-1}$, $r_0 = 2 \text{ a}_0$, $\alpha_r = 0.5 \text{ a}_0^{-1}$ und $m = 0.3 \text{ amu}$. Die Eigenfunktionen sind nicht normiert.

Die zeitliche Entwicklung des Systems (200) ist einerseits durch

$$\psi(t) = e^{-i\hat{H}t}\psi_0 \quad (201)$$

mit dem Hamiltonoperator (184) gegeben, andererseits lässt sich $\psi(t)$ — da es sich um Eigenfunktionen handelt — durch

$$\psi(t) = \frac{1}{\sqrt{2}} \left(e^{-iE_5t} |5\rangle + e^{-iE_6t} |6\rangle \right) \quad (202)$$

darstellen. Nachdem die Energien E_5 und E_6 nach (186) bekannt sind, kann somit die Genauigkeit des zur Berechnung von $\psi(t)$ verwendeten Verfahrens überprüft werden.

5.1.2 Diskretisierung durch DVR

Zur Diskretisierung der Wellenfunktion und des Hamiltonoperators wurde das in [41, 47] vorgeschlagene DVR-Verfahren angewandt, das im wesentlichen auf [6] zurückgeht und in Anhang B.2 hergeleitet wird.

Im vorliegenden Fall wird der betrachtete Bereich von r_{\min} bis r_{\max} gleichmäßig in D Punkte aufgeteilt. Die verwendeten Basisfunktionen sind die Eigenfunktionen des Kastenpotentials in der Form (270).

Wegen der Orthogonalität der verwendeten Funktionen (siehe Gl. (292), Anh. B.2) nimmt die Hamiltonmatrix folgende Form an (vgl. [6, Gl. (2.6)]):

$$H_{\alpha\beta} = T_{\alpha\beta} + V(r_\alpha)\delta_{\alpha\beta}. \quad (203)$$

Dabei ist die kinetische Energie durch (293) und das Potential durch (185) gegeben.

Die Wellenfunktion läßt sich diskretisieren und so zu einem endlichen Vektor umschreiben, wenn man jeweils den Wert der Wellenfunktion am betreffenden Gitterpunkt betrachtet:

$$(\psi_0)_\alpha = \psi_0(r_\alpha). \quad (204)$$

5.1.3 Propagationsverfahren

Basierend auf den vorangegangenen Überlegungen wurde ein FORTRAN-Programm entwickelt, das mit den gegebenen Daten die Ausgangswellenfunktion und die Hamiltonmatrix aufstellt und die Wellenfunktion propagiert. Nach einigen Vorversuchen und Abschätzungen stellte sich heraus, daß nur die in 4.2 vorgestellten Verfahren der Vektorpropagation sinnvoll einsetzbar sind, da eine $(D \times D)$ -Hamiltonmatrix bei feineren Unterteilungen und damit wachsendem D sehr groß wird und damit der Speicherbedarf für die komplexe Zeitentwicklungsmatrix sehr schnell die technischen Grenzen übersteigt, insbesondere bei Systemen mit mehreren Freiheitsgraden.

Als Vorgriff zu 5.3 sollen hier konkrete Zahlen genannt werden. Für das zweidimensionale HeI_2 -Problem wird in der Literatur (z. B. [41, 46]) üblicherweise mit einem 256×256 -Gitter gearbeitet. Dies hätte für die Matrix des Zeitentwicklungsoperators U bei einer Genauigkeit von DOUBLE COMPLEX einen Speicherbedarf von etwa 64 GB zur Folge, was mit Computern (zum gegenwärtigen Zeitpunkt noch) nicht sinnvoll zu bewältigen ist. Somit muß die Berechnung mit Vektormethoden nach 4.2 durchgeführt werden (ein komplexer Zustandsvektor mit 65536 Einträgen beansprucht etwa 1 MB Speicherplatz).

Die Hamiltonmatrix wurde in Einheiten von cm^{-1} berechnet. Setzt man wie in der Definition des Zeitentwicklungsoperators (3) $\hbar = 1$, so entspricht ein Zeitschritt von $\Delta t = 1$ einem Zeitintervall von

$$\tau = 5.31 \text{ ps} \hat{=} \frac{1}{1 \text{ cm}^{-1}}. \quad (205)$$

Die Grundfrequenz ω_0 des Jodidmoleküls beträgt unter diesen Konventionen nach (197) 128 cm^{-1} , was einer Periodendauer von

$$T = \frac{1}{\omega_0} = \frac{1}{128} \tau \approx 0.0415 \text{ ps} \quad (206)$$

entspricht.

Der Koordinatenbereich wurde mit

$$r_{\min} = 4.9, \quad r_{\max} = 6.7 \quad (207)$$

festgelegt (in Einheiten von a_0). Als Propagationsverfahren wurden das Tschebyscheff-Verfahren und die Beschleunigung mit dem vektoriellen ϵ -Algorithmus ausgewählt und jeweils mit dem exakten Propagationsergebnis nach (202) verglichen. Der Bereich der Eigenwerte wurde über die Maxima und Minima der Energien auf dem Gitter nach (68) unter Berücksichtigung der Nullpunktsenergie abgeschätzt.

Tabelle 8 zeigt die Anzahl der Gitterpunkte und die Anzahl der benötigten Approximationsschritte bei einer relativen Fehlerschranke von $g = 10^{-13}$.

Aus dieser Tabelle lassen sich mehrere Ergebnisse ableiten: zum einen steigt mit größerem D die maximal darstellbare kinetische Energie und damit die obere Grenze des Energiebereichs¹⁴; damit steigt auch das Δz in der Tschebyscheff-Entwicklung und wegen des exponentiellen Verhaltens der Besselfunktionen $J_n(\Delta z)$ für $n > \Delta z$ auch der Approximationsgrad. Im vorliegenden Fall resultiert daraus eine Erhöhung des Rechenaufwandes um einen Faktor 3–4 bei einer Verdoppelung der Anzahl der Punkte und ein zusätzlicher Faktor $2^2 = 4$ wegen der Verdoppelung der Vektordimension.

Das vektorielle ϵ -Verfahren zeigt ähnliches Verhalten, allerdings ist hier der Rechenaufwand im besten Fall etwa um einen Faktor 2.5 größer. Man sieht, daß grundsätzlich bei der Verringerung des Skalierungsgrades der Rechenaufwand etwas absinkt, um dann wieder stark anzusteigen. Bei noch größeren Normen (es wurden Berechnungen mit $\|Ht/s\|_1 \approx 50$ durchgeführt) ist keine Konvergenz zum richtigen Ergebnis mehr möglich, was ja aus den Voruntersuchungen bereits hervorgeht (siehe Abschnitt 4.1.3, Tabelle 2).

Für $D = 128$ Punkte ist eine Skalierung auf $\|Ht/s\|_1 \approx 30$ am besten, während für $D = 256$ ein größerer Skalierungsgrad s bessere Ergebnisse liefert. Der Grund für dieses unterschiedliche Verhalten liegt darin begründet, daß die 1-Norm $\|\cdot\|_1$ nur ein sehr oberflächliches Kriterium darstellt. Dies wird besonders in Spalte 4 deutlich, wo bei gleicher Norm ein sehr unterschiedliches Konvergenzverhalten zu beobachten ist. Bei Matrizen gleicher Norm können einzelne Einträge vom Betrag her sehr stark differieren und u. U. die interne Stellenzahl des Computers überschreiten. Um dieses Problem in den Griff zu bekommen, wäre es denkbar, den Wert des betragsgrößten Eintrags in der Matrix neben der 1-Norm zur Bestimmung des optimalen Skalierungsgrads s mit heranzuziehen, oder stattdessen die (rechenintensivere) 2-Norm zu verwenden.

Bei den Berechnungen zu Tabelle 8 stellte sich heraus, daß die Genauigkeit des ϵ -Verfahrens mit einem relativen Fehler von 0.1 % die der Tschebyscheff-Entwicklung (relativer Fehler etwa 1 %) um eine Größenordnung übertrifft. Zur Bestimmung des

¹⁴Die Wellenfunktionen $|5\rangle$ und $|6\rangle$ liegen sehr tief im Potentialtopf und sind im gewählten Raumbereich sehr gut lokalisiert; erst für Wellenfunktionen nahe bei der maximalen Schwingungszahl von $v_{\max} = 76$ macht sich das flache Auslaufen der Potentialkurve und das physikalisch „falsche“ Verhalten bei $r = 0$ bemerkbar. Für die vorliegende Wellenfunktion ist das energetische Minimum im wesentlichen durch $V(r_0) = -D_r$ und das Maximum wegen $V(r) \rightarrow 0$ für $r \rightarrow \infty$ hauptsächlich durch die maximal darstellbare kinetische Energie gegeben.

	$\frac{t}{T}$	$\ Ht\ _1$	Tschebyscheff		Vektor- ϵ			
			N		1	2	3	4
$D = 128$	1	350	267	682	640	624	795	
	2	700	492	1364	1262	1246	1602	
	3	1050	712	2064	1908	1879	2383	
	4	1399	930	2787	2522	2506	3402	
	5	1749	1147	3410	3168	3138	4145	
	6	2099	1364	4092	3782	3753	4974	
	7	2449	1579	4774	4428	4388	5974	
	8	2799	1794	5426	5042	4988	7068 [‡]	
	9	3149	2009	6107	5689	5613	7876 [†]	
	10	3499	2224	6789	6302	6256	8693 [†]	
$D = 256$	1	1452	852	2774	2571	2655	3565	
	5		3986	13853	12741	13311	17094 [‡]	
$D = 512$	1		3126	—	—	—	—	
	2		6151	—	—	—	—	

Der Wert für N gibt bei der Tschebyscheff-Approximation die Approximationsordnung an, während sich der Wert für das vektorielle ϵ -Verfahren aus der Gesamtzahl der Aufrufe der ϵ -Routine ergibt. In den Spalten 1–4 wurden unterschiedliche Skalierungen s gewählt, und zwar

im Fall 1: $\|Ht/s\|_1 \approx 16$

im Fall 2: $\|Ht/s\|_1 \approx 20$

im Fall 3: $\|Ht/s\|_1 \approx 30$

im Fall 4: $\|Ht/s\|_1 \approx 40$

Für s wurde diejenige ganze Zahl gewählt, für die die Norm der skalierten Matrix dem angegebenen Wert entsprach oder knapp darunter lag.

Zur Konvergenz der reskalierten Matrix Ht/s waren

im Fall 1 ca. 31,

im Fall 2 ca. 36,

im Fall 3 ca. 54 und

im Fall 4 88–101 Aufrufe notwendig.

Der relative Fehler, verglichen mit dem exakten Ergebnis nach Gl. (202), lag immer unter 0.1%, außer in den mit [†] bzw. [‡] markierten Fällen. Bei den mit [†] gekennzeichneten Ergebnissen lag der Fehler über 0.1%, aber immer noch unter 1%, während bei den mit dem [‡]-Symbol versehenen Resultaten aufgrund von Rundungsfehlern keine Konvergenz auf das exakte Ergebnis zu beobachten war.

Im Vergleich dazu ist der relative Fehler bei der Tschebyscheff-Approximation immer im Bereich 1%.

Für $D = 512$ Punkte wurde aus Zeitgründen nur die Tschebyscheff-Entwicklung herangezogen.

Tabelle 8: Ergebnisse der Zeitentwicklung von I_2 .

relativen Fehlers wurde der Bereich der Wellenfunktion herangezogen, in dem die Wahrscheinlichkeitsdichte im Rahmen der Zeichengenauigkeit von 0 verschieden ist. Als Beispiel ist in Abb. 14 der relative Fehler des Bertagsquadrates $|\psi(t)|$ zum Zeitpunkt $t = 9T$ dargestellt. Betrachtet wurde das Intervall von $\alpha = 20 \dots 110$. Hier ist das Wahrscheinlichkeitsquadrat von Null verschieden (vgl. Abb. 15) und der relative Fehler der Tschebyscheff-Entwicklung unter 10^{-2} und somit unter 1 %, der des ϵ -Verfahrens sogar unter 0.1 %. Das Ansteigen des relativen Fehlers auf 10^4 in der Nähe von $\alpha = 0$ ist für die Propagation nicht relevant, da die dort erreichten Werte bei 10^{-14} bzw. 10^{-18} und somit um 12 bis 16 Größenordnungen unter dem Maximum mit ca. 0.04 liegen.

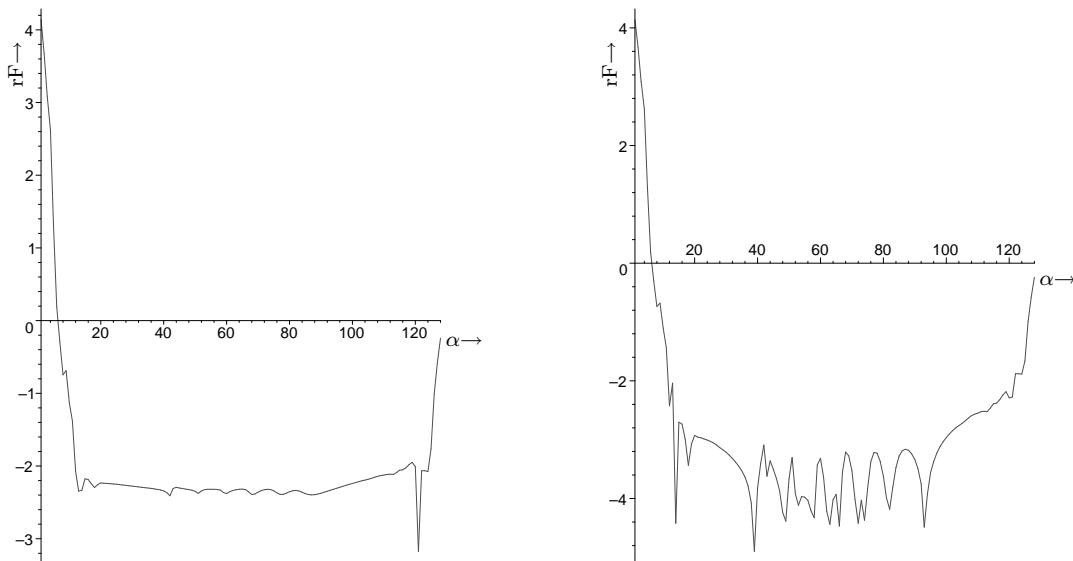


Abbildung 14: Logarithmus des relativen Fehlers von $|\psi(r_\alpha, t)|^2$, berechnet zum Zeitpunkt $t = 9T$ mit $D = 128$. Links wurde die Berechnung mit der Tschebyscheff-Approximation durchgeführt, rechts mit dem vektoriiellen ϵ -Verfahren bei einer Skalierung auf $\|Ht/s\|_1 \approx 30$. Man erkennt, daß der Fehler links in einem weiten Bereich unter 1 % liegt, rechts sogar unter 0.1 %. Das ϵ -Verfahren wäre somit um fast eine Größenordnung genauer, dafür ist der Rechenaufwand etwa 2.5 mal höher (vgl. Tab. 8). An den Rändern, wo der Fehler stark ansteigt, ist der Absolutbetrag der Wellenfunktion nahe bei Null.

Um das Verhalten des ϵ -Algorithmus noch genauer zu beleuchten, wurde die Berechnung zum Zeitpunkt $T = 9$ mit $D = 128$ Punkten und einer Skalierung auf $\|Ht/s\| \approx 30$ mit verschiedenen Fehlerschranken g pro Einzelpropagationsschritt durchgeführt. Tabelle 9 zeigt die Resultate.

Wie man erkennt, ist die Genauigkeit bei der Berechnung von $\psi(t)$ in weiten Bereichen unabhängig von der Fehlerschranke g (g wird hier immerhin um 7 Zehnerpotenzen verringert!). Gleichzeitig sinkt die Zahl der ϵ -Aufrufe, sodaß bei $g = 10^{-6}$ der Rechenaufwand gegenüber $g = 10^{-13}$ bei gleichbleibenden Resultaten um 40 % reduziert werden kann.

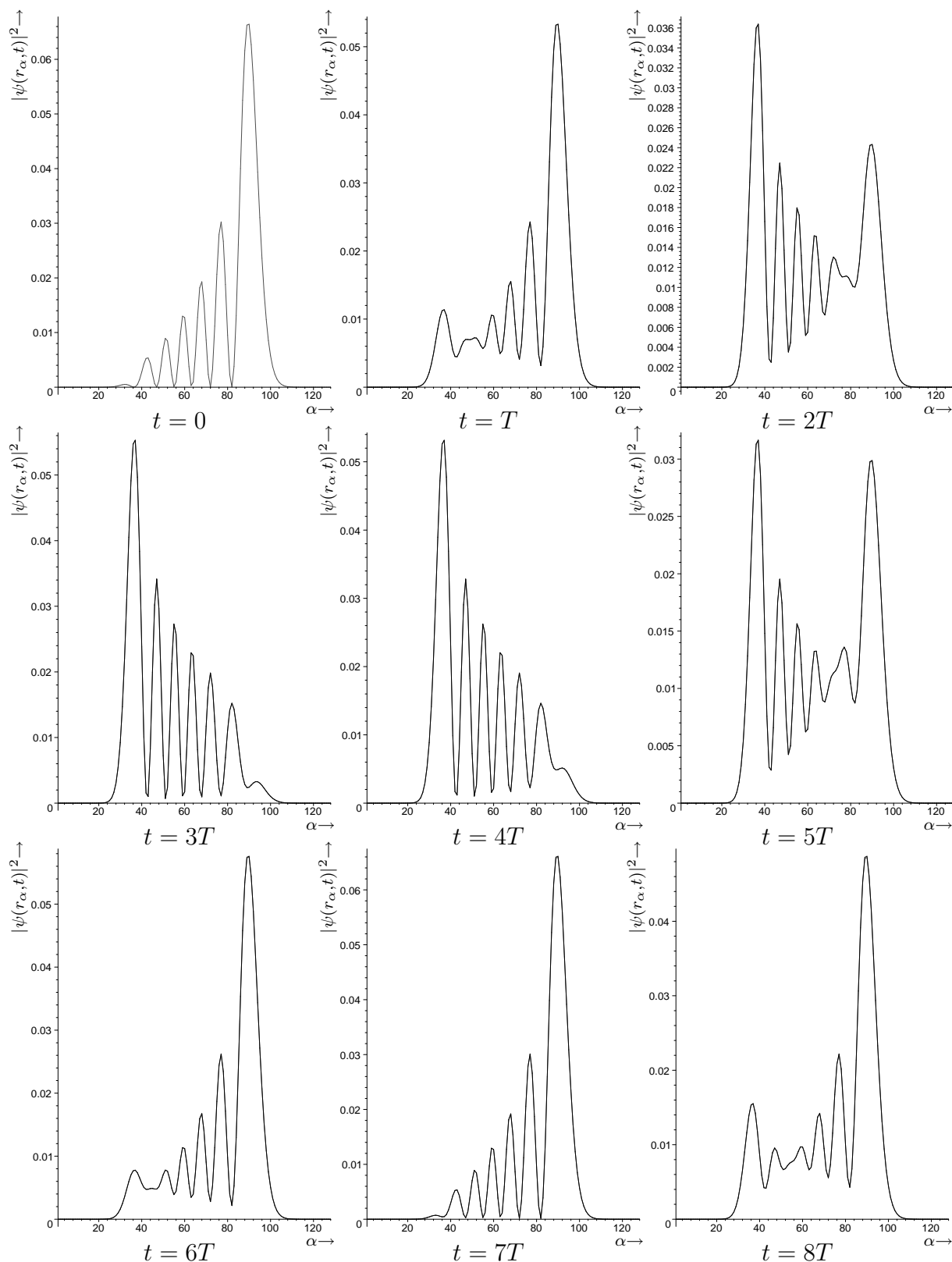
g	Anzahl der ϵ -Aufrufe	rF
10^{-13}	5613	10^{-3}
10^{-12}	5196	10^{-3}
10^{-11}	4801	10^{-3}
10^{-10}	4400	10^{-3}
10^{-9}	4095	10^{-3}
10^{-8}	3863	10^{-3}
10^{-7}	3683	10^{-3}
10^{-6}	3362	10^{-3}
10^{-5}	5091	≈ 1

Tabelle 9: Abhängigkeit der Anzahl der ϵ -Aufrufe und des relativen Fehlers rF von der Fehlerschranke g . Bis $g = 10^{-6}$ ist der ϵ -Algorithmus stabil, darunter treten Instabilitäten auf.

Somit ist das ϵ -Verfahren trotz des erhöhten Speicher- und Rechenbedarfs an Genauigkeit der Tschebyscheff-Approximation überlegen und stellt deshalb eine ernstzunehmende Alternative dar.

Um die Berechnungen zu illustrieren und die Zeitentwicklung von I_2 darzustellen, wurden in Abb. 15 die Ergebnisse der Tschebyscheff-Entwicklung für $D = 128$ und $t = 0 \dots 9T$ grafisch dargestellt. In den gezeigten Abbildungen wurde jeweils (bis auf den Fall $t = 0$) die aus der Approximation erhaltene Lösung mit der exakten in ein Diagramm gezeichnet. Man sieht im Rahmen der Zeichengenauigkeit völlige Übereinstimmung der beiden Kurven. Somit ist zumindest qualitativ die Verwendbarkeit der getesteten Verfahren bestätigt.

Detailliertere Untersuchungen (z. B. mit extrem großen Zeitschritten oder feineren Auflösungen) würden den Rahmen dieses Kapitels sprengen, das — wie zu Anfang bereits angedeutet — als Einführung in die Methodik und als Vorbereitung auf das übernächste Kapitel zu sehen ist.

Abbildung 15: Zeitentwicklung von I_2 (Fortsetzung auf der nächsten Seite).

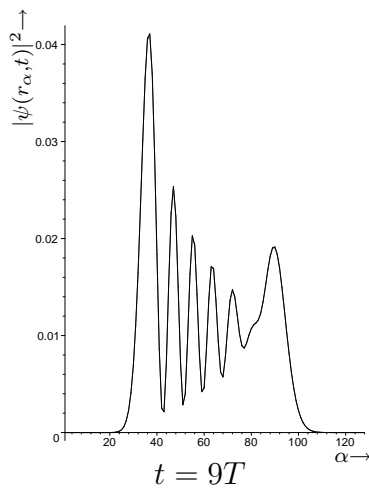


Abbildung 15 (Fortsetzung): Zeitentwicklung von I_2 in Zeitabständen von $T = \frac{1}{\omega_0} = 0.0415$ ps. Der Einfachheit halber wurde die Wahrscheinlichkeitsdichte $|\psi(r_\alpha, t)|^2$ als Funktion des Index α angegeben und auf eine Umrechnung in Abstände gemäß (269) verzichtet.

5.2 Propagation durch SPO-Verfahren

Als direkter Vergleich wurde die Zeitentwicklung von I_2 mit dem in Abschnitt 3.4 vorgestellten split operator-Verfahren durchgeführt und ebenso wie im vorigen Abschnitt mit der exakten Lösung nach Gl. (202) verglichen.

Für die durchzuführenden Fast Fourier-Transformationen wurde die Routine `C06FCF` aus der NAG-Bibliothek verwendet [38]. Diese Routine berechnet aus einer Reihe z_α von komplexen Zahlen ($\alpha = 0, \dots, D-1$) die Fourier-transformierten Größen \hat{z}_β ($\beta = 0, \dots, D-1$) nach

$$\hat{z}_\beta = \frac{1}{\sqrt{D}} \sum_{\alpha=0}^{D-1} z_\alpha \exp\left(-2\pi i \frac{\alpha\beta}{D}\right). \quad (208)$$

Diese Transformation wird verwendet, um die Wellenfunktion von der Orts- in die Impulsdarstellung überzuführen. Dazu werden für die z_α die Werte

$$z_\alpha = \psi(r_\alpha)$$

der Wellenfunktion ψ an den Gitterpunkten α eingesetzt; die Größen \hat{z}_β bilden dann die Transformierte, also

$$\hat{z}_\beta = \hat{\psi}(k_\beta).$$

Den Punkten r_α im Ortsraum werden Impulse k_α zugeordnet gemäß¹⁵

$$k_\alpha = \left(\alpha - \frac{D}{2}\right) \frac{2\pi}{D(r_{\max} - r_{\min})}. \quad (209)$$

Der Betrag des Impulses liegt somit im Bereich

$$0 \leq |k| \leq \frac{\pi}{r_{\max} - r_{\min}}. \quad (210)$$

Bei der Propagation wurde der vorgegebene Zeitschritt Δt in N kleinere Zeitschritte δt zerlegt, sodaß zum einen die Gleichung

$$\Delta t = N\delta t \quad (211)$$

und zum anderen das Stabilitätskriterium

$$\delta t \approx \frac{\Delta t_{\text{krit}}}{5} \quad (212)$$

erfüllt ist. Δt_{krit} wurde in Abschnitt 3.4, Gl. (143) definiert. Somit gilt:

$$N = 5 \left\lceil \frac{\Delta t}{\Delta t_{\text{krit}}} \right\rceil. \quad (213)$$

¹⁵Gl. (209) setzt voraus, daß D gerade ist. Für die effiziente Verwendung von FFT-Algorithmen ist es empfehlenswert, daß D eine Potenz von 2 ist [11], sodaß diese Voraussetzung in der Praxis keine Einschränkung darstellt.

Die Klammern $[x]$ bezeichnen hierbei den ganzzahligen Anteil von x . Zur Propagation über den Zeitschritt Δt müssen also N Transformationen bzw. Rücktransformationen durchgeführt werden.

Zur Berechnung der Rücktransformation in den Ortsraum wurde die Zahlenfolge zuerst komplex konjugiert, dann mit der Routine C06FCF transformiert und anschließend wieder komplex konjugiert. Dies entspricht einer inversen Transformation. Somit sind $2N$ Transformationsaufrufe notwendig.

Tabelle 10 zeigt den Rechenaufwand bei einer Unterteilung des Raumbereichs in $D = 128$ bzw. $D = 256$ Gitterpunkte.

$\Delta t/10^{-3}$ ps	Zahl der Transformationen N	
	$D = 128$	$D = 256$
1	37	107
2	75	215
3	113	323
4	151	431
5	189	539
6	226	647
7	264	755
8	302	863
9	340	971
10	378	1079
10^2	3780	10798
10^3	37801	107984

Tabelle 10: Anzahl der Transformationen N bei gegebenem Zeitschritt Δt . Der Raumbereich wurde in $D = 128$ bzw. $D = 256$ Punkte aufgeteilt.

Anhand der vorliegenden Daten¹⁶ konnte die theoretisch geforderte Skalierung des Rechenaufwandes mit $D \log_2 D$ überprüft werden¹⁷. Hierzu wurde mit der Kompileroption

¹⁶Es zeigte sich, daß die Berechnungen bis etwa $\Delta t = 10^{-2}$ ps stabil sind, ab dann werden Abweichungen der Graphen (exakte Lösung/SPO-Ergebnis) voneinander erkennbar, die mit steigendem Δt zunehmen. Dieses Verhalten ist unabhängig davon, mit welcher Punktauflösung propagiert wird. Auch eine Verschärfung des Stabilitätskriteriums auf

$$\delta t \approx \frac{\Delta t_{\text{krit}}}{20} \quad (214)$$

brachte hier keine Veränderungen. Somit liegt den Berechnungen ein systematischer Fehler zugrunde, der aus Zeitgründen leider nicht mehr korrigiert werden konnte. Dies betrifft aber nicht die folgenden Ausführungen bezüglich des Rechenaufwandes, sodaß dieses Kapitel trotzdem (mit diesen Einschränkungen) in die Arbeit aufgenommen wurde. Aussagen über die Genauigkeit können allerdings — abgesehen von der theoretisch geforderten $O(\Delta t^3)$ -Abhängigkeit des Fehlers — nicht gemacht werden.

¹⁷In [38] wird darauf hingewiesen, daß diese theoretische Grenze nur mit Potenzen von 2 erreicht werden kann. Deshalb wurden hier die Werte $D = 128 = 2^7$ und $D = 256 = 2^8$ verwendet.

-pg der Profiler aktiviert, der u. a. die Laufzeit des Programms protokollierte. Um verlässliche Daten zu erhalten, wurden größere Propagationsschritte bis zu $\Delta t = 1$ ps in die Tabelle 10 aufgenommen.

Es ergaben sich folgende Laufzeiten:

$\Delta t/10^{-3}$ ps	Laufzeit t_{total} in s	
	$D = 128$	$D = 256$
10^2	2.92	15.66
10^3	26.68	156.56

Um den Offset t_{off} durch Initialisierungsroutinen u. ä. zu eliminieren, wurde folgendes Gleichungssystem zur Bestimmung der Zeit t_{step} für einen Propagationsschritt herangezogen:

$$\begin{aligned} N_1 t_{\text{step}} + t_{\text{off}} &= t_{\text{total},1} \\ N_2 t_{\text{step}} + t_{\text{off}} &= t_{\text{total},2} \end{aligned}$$

Setzt man hier die Werte für N und die Laufzeiten ein, so erhält man:

$$\begin{aligned} t_{\text{step}} &= 7.0 \cdot 10^{-4} \text{ s} && \text{für } D = 128, \\ t_{\text{step}} &= 1.5 \cdot 10^{-3} \text{ s} && \text{für } D = 256. \end{aligned}$$

Für das Verhältnis des Rechenaufwandes ergibt sich hier

$$\frac{t_{\text{step}, D=256}}{t_{\text{step}, D=128}} = \frac{15}{7} \approx 2.14, \quad (215)$$

während theoretisch

$$\frac{256 \log_2 256}{128 \log_2 128} = \frac{16}{7} \approx 2.29 \quad (216)$$

zu erwarten ist. Die beiden Werte stimmen relativ gut überein, die Abweichungen ergeben sich hauptsächlich daraus, daß der Profiler selbst die Laufzeit der Programme beeinflusst.

Somit ist das SPO-Verfahren gerade bei größeren Problemen gut geeignet, den Rechenaufwand zu minimieren. Insbesondere bei dem in Abschnitt 5.3 behandelten Problem ließen sich die Rechnungen deutlich beschleunigen. Wie bereits festgestellt, kann mit dem hier verwendeten Algorithmus jedoch nur über Zeiten in der Größenordnung $\Delta t = 10^{-2}$ ps propagiert werden, was für die dort anfallenden Berechnungen nicht ausreichend ist.

5.3 Das van der Waals-Molekül HeI_2

5.3.1 Physikalische Grundlagen

In [42] ist ein T-förmiges Modell von HeI_2 ohne Rotation beschrieben. Dies ist ein Drei-Körper-Problem mit dem Hamilton-Operator

$$\hat{H} = -\frac{1}{2m} \frac{\partial^2}{\partial r^2} - \frac{1}{2\mu} \frac{\partial^2}{\partial R^2} + \hat{V}_{\text{I-I}}(r) + 2\hat{V}_{\text{He-I}}(\rho(r, R)), \quad (217)$$

wobei die potentiellen Energien jeweils wieder durch Morse-Potentiale der Form

$$V_{\text{I-I}}(r) = D_r \left[e^{-2\alpha_r(r - r_0)} - 2e^{-\alpha_r(r - r_0)} \right] \quad (218)$$

bzw.

$$V_{\text{He-I}}(\rho) = D_\rho \left[e^{-2\alpha_\rho(\rho - \rho_0)} - 2e^{-\alpha_\rho(\rho - \rho_0)} \right] \quad (219)$$

dargestellt werden. Dabei ist R der Abstand des Helium-Atoms vom Schwerpunkt des I_2 , r der Abstand der beiden Jod-Atome und

$$\rho = \rho(r, R) = \sqrt{R^2 + r^2/4} \quad (220)$$

der Abstand des Helium-Atoms zu einem Jod-Atom (siehe auch Abb. 16). Dabei wird die Bewegung des Heliumatoms stets senkrecht zur Verbindungslinie der Jod-Atome angenommen, d. h. das System besitzt nur zwei Freiheitsgrade. μ ist die reduzierte Masse von He-I und m die reduzierte Masse des I_2 . Dieses Modell ist von Interesse

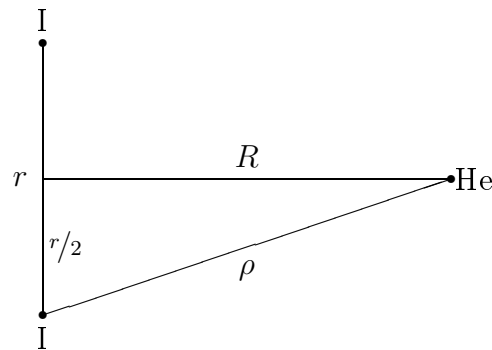


Abbildung 16: Zur Geometrie des HeI₂-Problems.

hinsichtlich einer unimolekularen Dissoziation der Form



und ist theoretisch von Gray, Rice und Mitarbeitern [42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58] als auch experimentell [59, 60] gut untersucht, so daß Vergleichsdaten vorliegen.

In [42] wird beschrieben, daß die klassische Bewegung eines Systems aus gekoppelten, nichtlinearen Oszillatoren bei geringen Energien meist quasiperiodisch verläuft, während bei höheren Energien die Bewegung chaotisch wird. Eine chaotische Bewegung erlaubt einen sehr schnellen intramolekularen Energietransfer, sodaß die Dissoziationsrate prinzipiell durch einen statistischen Formalismus berechnet werden kann. Studien von Oxtoby und Rice [61] dagegen zeigen, daß isolierte nichtlineare Resonanzen auftreten können, wenn die Frequenzen der gekoppelten Oszillatoren stark voneinander

abweichen. Diese Resonanzen können Energie speichern und beeinflussen die Dynamik des Systems beträchtlich, sodaß diese sich einer rein statistischen Beschreibung entzieht.

Als wichtiges Ergebnis stellen Gray, Rice und Noid in [42] fest, daß für niedrige Vibrationszustände ($v \leq 5$) des I_2 -Anteils die Dissoziation von HeI_2 rein quantenmechanisch begründet ist (unter klassischer Beschreibung wäre dieses Molekül also stabil), während für höhere Zustände die Dissoziation auch klassisch erlaubt ist und hier sowohl quantenmechanische als auch klassische Faktoren die Dissoziationsrate beeinflussen. Experimentell wurden die Ergebnisse überprüft, indem Helium unter hohem Druck über festes Jod bei Zimmertemperatur geführt wurde und dann durch eine kleine Öffnung in einen Bereich mit niedrigem Druck expandieren konnte [59, 60]. Der so entstehende Überschallstrahl wurde mit einem durchstimmbaren Farbstofflaser angeregt und spektroskopisch untersucht. Es zeigte sich hier, daß die Dissoziation von HeI_2 hauptsächlich durch den Übergang $v \rightarrow v'$ mit $v' = v - 1$ bestimmt wird.

Die verwendeten Potentialdaten von Jod wurden bereits an anderer Stelle verwendet. Hier sind sie noch einmal zum Vergleich zusammen mit den Daten für das He-I-Potential angegeben (aus [41]):

$$\begin{aligned} D_r &= 4911 \text{ cm}^{-1}, & \alpha_r &= 0.9380 \text{ a}_0^{-1}, & r_0 &= 5.6994 \text{ a}_0, \\ D_\rho &= 18 \text{ cm}^{-1}, & \alpha_\rho &= 0.6033 \text{ a}_0^{-1}, & \rho_0 &= 7.5589 \text{ a}_0. \end{aligned} \quad (222)$$

Man sieht anhand dieser Daten die unterschiedliche Bindungsstärke zwischen den beiden Jodatomen (D_r) und der van der Waals-Bindung (D_ρ). Der Potentialverlauf ist in Abb. 17 grafisch dargestellt.

In [41] wurde mit Hilfe des DVR-Verfahrens der quasi-gebundene Anfangszustand numerisch ermittelt und dieser unter Anwendung der split operator-Methode (s. 3.4 und 5.2) propagiert. Hier soll ein anderer Weg besprochen werden: es zeigt sich (siehe 5.3.3), daß sich in guter Näherung die Bewegung der Jodatome entlang der r -Koordinate von der des Heliumatoms (R) entkoppeln läßt, sodaß der Anfangszustand ψ_0 als Produkt zweier Eigenzustände der jeweiligen Morsepotentiale beschreiben werden kann. Diese Näherung wird dann aufgehoben, um die Wechselwirkung zwischen Jod und Helium zu ermöglichen. Der Hamiltonoperator (217) wird dann durch Wahl eines Gitters und geeigneter Basisfunktionen der Form (270) mittels DVR-Formalismus in eine Matrix übersetzt (siehe 5.3.2), um die Wellenfunktion ψ_0 propagieren zu können. Diese Vorgehensweise stützt sich auf [42, 46, 47, 56].

5.3.2 Diskretisierung durch DVR

Die Diskretisierung wurde analog zu 5.1.2 durchgeführt; deshalb werden hier im wesentlichen die Unterschiede herausgestellt, die sich durch den zusätzlichen Freiheitsgrad ergeben. Der Übersichtlichkeit wegen werden einige Ergebnisse aus dem oben zitierten Kapitel nochmals aufgeführt.

Da es sich bei den hier betrachteten Problem um ein System mit zwei Freiheitsgraden (r , R) handelt, wird ein zweidimensionales Gitter gewählt, das in r -Richtung den

Bereich von r_{\min} bis r_{\max} und in R -Richtung den Bereich von R_{\min} bis R_{\max} abdeckt. Die Anzahl der Punkte in r - und R -Richtung betrage gleichermaßen D . Im folgenden bezeichnet der Index α die r -Richtung, der Index β die R -Richtung.

Die verwendeten Basisfunktionen sind die bereits in 5.1 verwendeten Eigenfunktionen des Kastenpotentials in der Form

$$\langle r_\alpha | j \rangle = \sqrt{\frac{1}{D+1}} \sin\left(\frac{\alpha j \pi}{D+1}\right) \quad (223)$$

bzw.

$$\langle R_\beta | k \rangle = \sqrt{\frac{1}{D+1}} \sin\left(\frac{\beta k \pi}{D+1}\right). \quad (224)$$

Der Indizes j und k numerieren die Eigenfunktionen durch und laufen jeweils von 1 bis D . Die r_α , R_β sind dabei analog zu Gln. (268), (269) in Anhang B.2 definiert.

Aus diesen Funktionen werden Produkte der Form $\langle r_\alpha R_\beta | jk \rangle$ gebildet. Im zweidimensionalen Fall nimmt die Hamiltonmatrix unter Berücksichtigung der Orthogonalität der Basisfunktionen die folgende Form an (vgl. (203)):

$$\mathbf{H}_{\alpha\beta, \alpha'\beta'} = \mathbf{T}_{\alpha\alpha'} \delta_{\beta\beta'} + \mathbf{T}_{\beta\beta'} \delta_{\alpha\alpha'} + V_{I-1}(r_\alpha) \delta_{\alpha\alpha'} \delta_{\beta\beta'} + V_{\text{He-1}}(\rho(r_\alpha, R_\alpha)) \delta_{\alpha\alpha'} \delta_{\beta\beta'}. \quad (225)$$

Dabei sind die kinetischen Energien durch

$$T_{\alpha\alpha'} = -\frac{1}{2m} \left\langle r_\alpha \left| \frac{\partial^2}{\partial r_{\alpha'}^2} \right| r_{\alpha'} \right\rangle \quad (226)$$

bzw.

$$T_{\beta\beta'} = -\frac{1}{2\mu} \left\langle R_\beta \left| \frac{\partial^2}{\partial R_{\beta'}^2} \right| R_{\beta'} \right\rangle \quad (227)$$

und die Potentiale durch (185) und (219) gegeben.

Damit \mathbf{H} in „klassischer“ Weise als Matrix dargestellt werden kann, wird eine eindeutige Abbildung des Doppelindex $\alpha\beta$ auf einen einfachen Index γ definiert durch¹⁸:

$$\gamma : (\alpha, \beta) \rightarrow (\beta - 1)D + \alpha, \quad \alpha = 1, \dots, D, \quad \beta = 1, \dots, D \quad (228)$$

Die Ausgangswellenfunktion ψ_0 läßt sich diskretisieren und als Vektor darstellen mit den Komponenten

$$(\psi_0)_\gamma = (\psi_0)_{\alpha\beta} = \psi_0(r_\alpha, R_\beta) \quad (229)$$

¹⁸natürlich wäre auch $\tilde{\gamma} : (\alpha, \beta) \rightarrow (\alpha - 1)D + \beta$ möglich.

5.3.3 Die Wellenfunktion ψ_0

Um ψ_0 in analytischer Form darstellen zu können, ist eine Entkopplung der beiden Koordinaten (r, R) notwendig. Nachdem die Kopplung über die Koordinate $\rho = \rho(r, R)$ zustandekommt, betrachtet man die Taylorentwicklung von $\rho(r, R)$ (vgl. dazu auch [42]):

$$\begin{aligned}\rho(r, R) &\approx \rho_0 + (r - r_0) \frac{\partial \rho}{\partial r}(r_0, R_0) + (R - R_0) \frac{\partial \rho}{\partial R}(r_0, R_0) \\ &= \rho_0 + \frac{1}{4}(r - r_0) \frac{r_0}{\rho_0} + (R - R_0) \frac{R_0}{\rho_0}\end{aligned}\quad (230)$$

Einsetzen von ρ_0 , r_0 und $R_0 = \sqrt{\rho_0^2 - r_0^2/4}$ aus (222) liefert:

$$\frac{R_0}{\rho_0} \approx 5 \cdot \frac{1}{4} \frac{r_0}{\rho_0}, \quad (231)$$

sodaß näherungsweise die Abhängigkeit von r vernachlässigt werden kann:

$$\tilde{\rho} = \tilde{\rho}(R) = \rho_0 + (R - R_0) \frac{R_0}{\rho_0}. \quad (232)$$

Setzt man nun $\tilde{\rho}$ anstelle von ρ in das Potential $V_{\text{He-I}}$ ein, erhält man:

$$2V_{\text{He-I}}(\tilde{\rho}) = 2D_\rho \left[e^{-2\alpha_\rho(\tilde{\rho} - \rho_0)} - 2e^{-\alpha_\rho(\tilde{\rho} - \rho_0)} \right] \quad (233)$$

$$= D_R \left[e^{-2\alpha_R(R - R_0)} - 2e^{-\alpha_R(R - R_0)} \right] \quad (234)$$

$$= V_R(R) \quad (235)$$

unter Ausnutzung von (232) und mit den Definitionen

$$\alpha_R = \frac{R_0}{\rho_0} \alpha_\rho = 0.5588 \text{ a}_0^{-1}, \quad (236)$$

$$D_R = 2D_\rho = 36 \text{ cm}^{-1}, \quad (237)$$

$$R_0 = \sqrt{\rho_0^2 - \frac{r_0^2}{4}} = 7.0012 \text{ a}_0. \quad (238)$$

Der Hamiltonoperator (217) läßt sich mit dem Potential V_R umschreiben zu

$$\begin{aligned}\hat{H} &= \underbrace{-\frac{1}{2m} \frac{\partial^2}{\partial r^2} + \hat{V}_{\text{I-I}}(r) - \frac{1}{2\mu} \frac{\partial^2}{\partial R^2} + \hat{V}_R(R)}_{\hat{H}_0} + \underbrace{2\hat{V}_{\text{He-I}}(\rho(R, r)) - \hat{V}_R(R)}_{\hat{V}} \\ &= \hat{H}_0 + \hat{V}\end{aligned}\quad (239)$$

Vernachlässigt man aufgrund obiger Überlegungen den Anteil \hat{V} , so zerfällt $\hat{H} \approx \hat{H}_0$ in zwei voneinander unabhängige Hamiltonoperatoren \hat{H}_r bzw. \hat{H}_R , sodaß sich die Eigenfunktion ψ_0 als Produkt zweier Eigenfunktionen zum Morsepotential schreiben läßt.

Setzt man die im vorliegenden Fall zugrundeliegenden Zahlenwerte ein, so erhält man

$$\begin{aligned} \kappa_r = 153.5 &\Rightarrow 0 \leq v_r \leq 76 \\ \kappa_R = 5.450 &\Rightarrow 0 \leq v_R \leq 2. \end{aligned} \quad (240)$$

5.3.4 Wahl des Koordinatenbereiches

Der maximale Koordinatenbereich wurde in Anlehnung an [41] mit

$$\begin{aligned} r_{\min} = 4.5, \quad r_{\max} = 8.0, \\ R_{\min} = -4.0, \quad R_{\max} = 60 \end{aligned} \quad (241)$$

festgelegt (in Einheiten von a_0). Das Potential $V_{\text{ges}}(r, R)$ in diesem Bereich, das nach (217) gemäß

$$V_{\text{ges}}(r, R) = V_{\text{I-I}}(r) + 2V_{\text{He-I}}(\rho(r, R)) \quad (242)$$

gegeben ist, ist in Abb. 17 als Höhenliniendiagramm abgebildet. Die eingetragenen Potentialwerte sind in Einheiten von 10^3 cm^{-1} , die Abstände in a_0 angegeben. Der

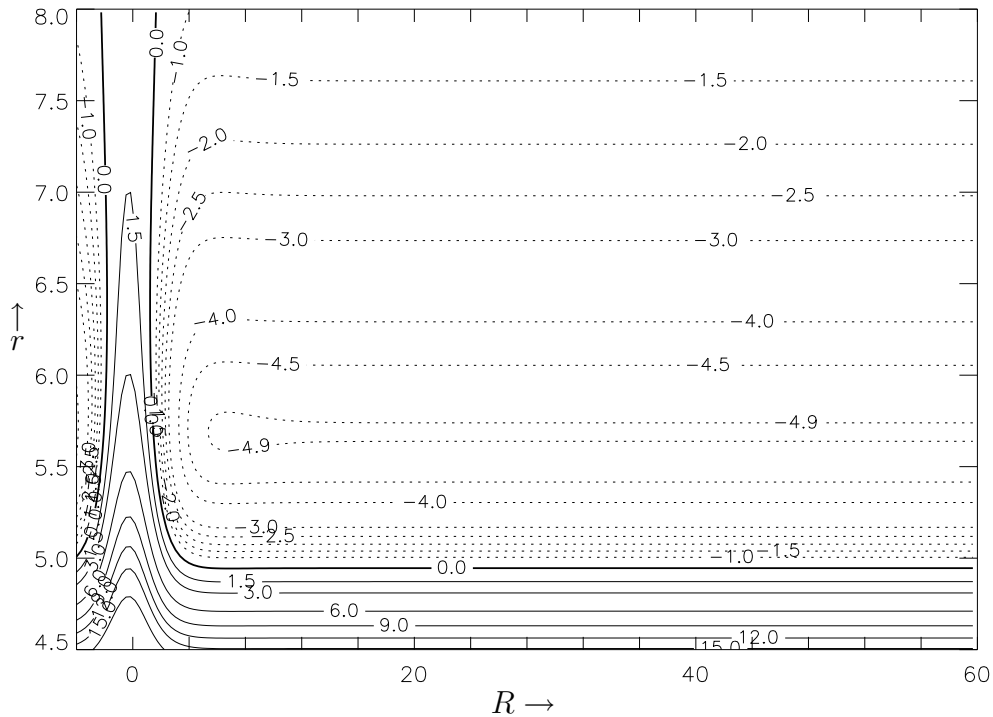


Abbildung 17: Potentialverlauf für das HeI_2 -Problem in den Koordinaten r und R

negative Wert für R_{\min} soll Randeinflüsse minimieren; Anteile der Wellenfunktion, die durch die endliche Potentialschwelle bei $R = 0$ tunneln und am linken Rand des Gitters reflektiert werden, müssen auf ihrem Weg zurück diese Barriere nochmals überwinden und werden dadurch sehr stark abgeschwächt. Somit hat der physikalisch falsche endliche Wert des Potentials an den Stellen $R = 0$ keinen nennenswerten praktischen Einfluß (vgl. dazu Fußnote 11 auf Seite 56).

R_{\max} ist sehr groß gewählt, damit die Wellenfunktion über genügend große Zeitschritte propagiert werden kann, ohne daß Einflüsse vom rechten Rand berücksichtigt werden müssen (vgl. dazu auch die Anmerkungen im Anhang von [46])¹⁹.

Nachdem der Jodid-Anteil des Moleküls nicht dissoziiert, sind Einflüsse von den Rändern des r -Bereichs nicht zu erwarten. Die Wellenfunktion ist für die betrachteten Jod-Vibrationszustände (im unteren Teil des Potentials) gut lokalisiert und fällt nach außen exponentiell ab (siehe dazu auch die Ergebnisse aus 5.1). In [41] wird beispielsweise die Wellenfunktion mit $v_r = 20$, $v_R = 0$ propagiert²⁰. Diese besitzt nach Gleichung (186) die Energie $E_{20,0} \approx -2661 \text{ cm}^{-1}$. Aus Abb. 17 ist ersichtlich, daß diese Wellenfunktion im wesentlichen innerhalb der gestrichelten Linie für $V = -2.5 \cdot 10^3 \text{ cm}^{-1}$ lokalisiert ist, d.h. im Bereich $5.1 \leq r \leq 7.0$.

Die Auflösung des Gitters sollte an die Zahl der Knoten der Wellenfunktion angepaßt werden; nach [6] reichen drei bis vier Gitterpunkte pro de Broglie-Wellenlänge für eine relative Genauigkeit von 10^{-4} bis 10^{-5} aus.

5.4 Propagation und Ergebnisse

Zuerst wurde der Zustand mit $v_r = 5$ auf einem 50×50 -Gitter berechnet und mit dem Tschebyscheff-Verfahren propagiert. Das Maximum R_{\max} wurde im Gegensatz zu (241) auf $7.0 a_0$ reduziert, um die relative Auflösung zu erhöhen. Abbildung 18 zeigt das Betragsquadrat der Wellenfunktion zum Zeitpunkt $t = 0$. Zur Darstellung wurde ein Konturplot gewählt, der mit IDL 5.0 [62] aus den berechneten Daten erzeugt wurde. Die Darstellung ist logarithmisch; der Übergang von einer Konturlinie zur nächsten entspricht zum Zentrum der Wellenfunktion hin einem Anwachsen der Wahrscheinlichkeitsdichte um einen Faktor 10.

Der Zeitschritt wurde mit $\Delta t = 0.1 \text{ ps}$ festgelegt. Für die Propagation über diesen Zeitschritt waren 462 Approximationsschritte notwendig. Die Grafiken in Abb. 20 zeigen das Betragsquadrat der Wellenfunktion zu den Zeiten $t = 1.0 \text{ ps}$, $t = 2.0 \text{ ps}$ und $t = 3.0 \text{ ps}$. Man sieht deutlich, daß sich die Aufenthaltswahrscheinlichkeit mit steigendem t in Bereiche mit größerem R ausdehnt. Das bedeutet, daß sich das Helium-

¹⁹Nicht bei allen physikalischen Problemstellungen ist es möglich, die Randbedingungen so zu wählen, daß Randeinflüsse umgangen werden können (man denke z. B. an ein Stoßproblem mit zwei Reaktionspartnern, deren Geschwindigkeiten sich nach dem Stoß sehr stark unterscheiden. Hier könnte die auslaufende Welle des einen Teilchens am Rand reflektiert werden und die Welle des anderen Teilchens überlagern und die Ergebnisse verfälschen). Eine Möglichkeit, dies zu vermeiden, liefert die Verwendung von sog. *optischen Potentialen*. Dies wird in Anhang C genauer ausgeführt.

²⁰Die He-I-Verbindung soll im Ausgangszustand nicht angeregt sein, d. h., es wird im folgenden immer implizit $v_R = 0$ vorausgesetzt.

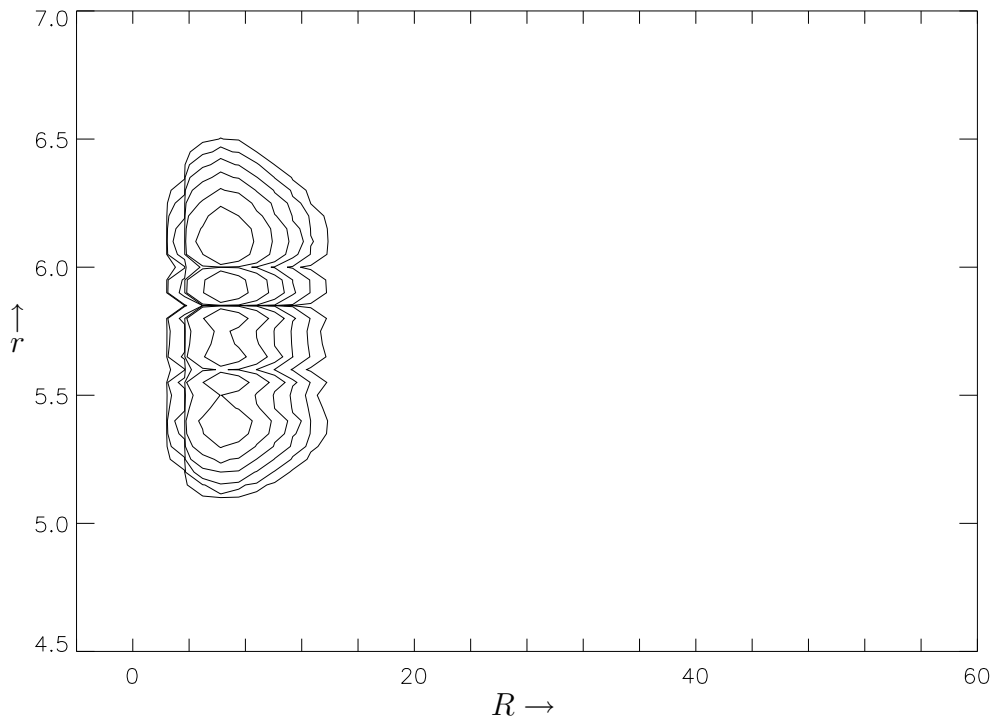


Abbildung 18: $|\psi_0|^2$ für $v_r = 5$; Gittergröße 50×50 Punkte

Atom zu späteren Zeiten mit steigender Wahrscheinlichkeit in größerer Entfernung vom Jodid-Anteil aufhält.

Bei den Berechnungen zeigte sich außerdem ein oszillatorisches Verhalten in r -Richtung. Um dies zu illustrieren und die Zeitentwicklung anschaulicher darzustellen, wurde aus den einzelnen Momentaufnahmen der Wellenfunktion im Abstand $t = 0.1$ ps eine Sequenz erstellt, die die Propagation von HeI_2 mit $v_r = 5$ im Zeitintervall 0 bis 3.5 ps zeigt²¹. Diese Animation ist auf der beiliegenden Diskette im Verzeichnis /animationen unter `v5_anim.gif` zu finden.

Um den Zusammenhang zwischen Gittergröße bzw. Auflösung und Rechenaufwand etwas genauer zu beleuchten und das Verhalten der Wellenfunktion für höhere Quantenzahlen v_r zu untersuchen, wurde noch der Zustand mit $v_r = 10$ auf einem 100×100 -Gitter analog zu obigen Ausführungen propagiert, diesmal allerdings auf dem vollen Koordinatenbereich nach (241). Abbildung 19 zeigt die Ausgangssituation.

Als Zeitschritt wurde $\Delta t = 0.05$ ps gewählt. Die Zahl der Approximationsschritte pro Zeitschritt betrug in diesem Fall 279. Die Berechnung eines Zeitschrittes betrug auf der ZEUS (SGI-Challenge Vektorrechner) immerhin 14.5 Stunden.

Abbildung 21 zeigt das Absolutquadrat der Wellenfunktion zu den Zeiten $t = 0.5$ ps,

²¹Die Animation wurde mit dem Programm WhirlGIF Version 2.01 von Kevin Kadow erzeugt. Die Software ist frei erhältlich unter <http://www.msg.net/utility/whirlgif/>.

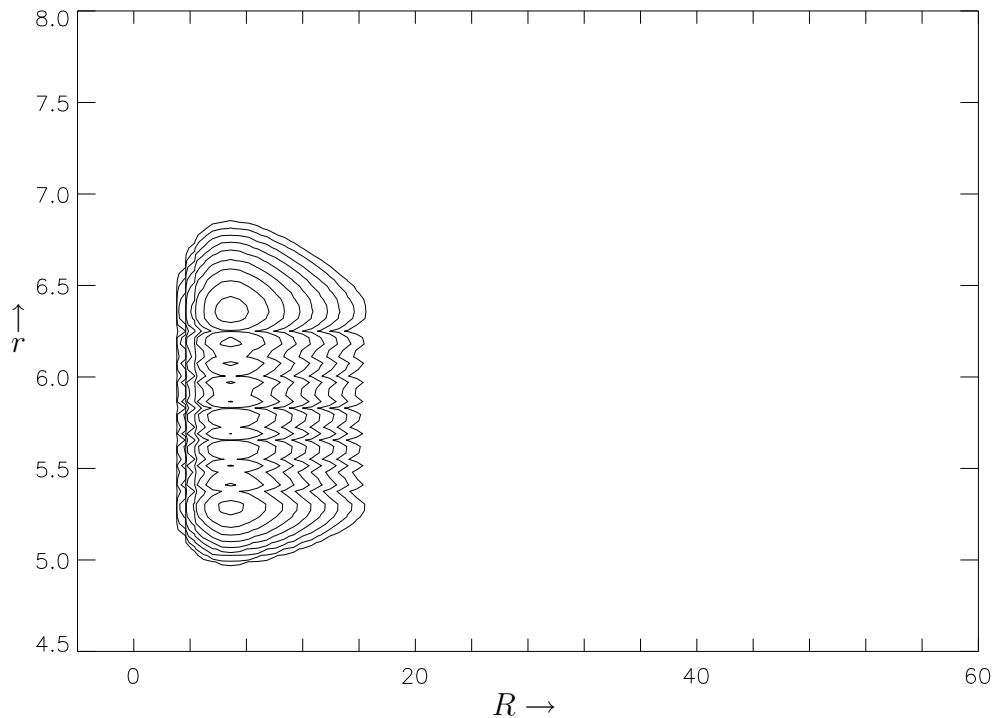


Abbildung 19: $|\psi_0|^2$ für $v_r = 10$; Gittergröße 100×100 Punkte

$t = 1.0$ ps und $t = 1.5$ ps.

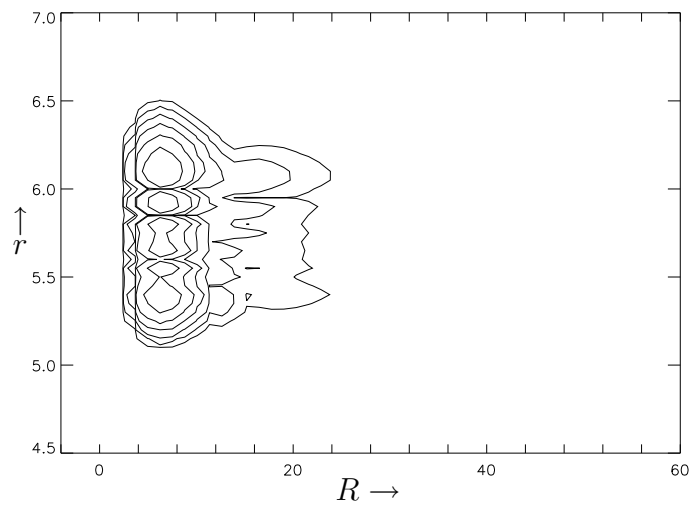
Auch hier wurde eine kleine animierte Sequenz erstellt; sie ist im gleichen Verzeichnis unter `v10_anim.gif` zu finden. Die Animation zeigt die Zeitentwicklung von $t = 0$ bis $t = 1.5$ ps in Schritten von $\Delta t = 0.05$ ps.

Man erkennt an den Momentaufnahmen in Abb. 21 und anhand der animierten Bildfolge ein grundsätzlich ähnliches Verhalten der Wellenfunktion im Vergleich zu Abb. 20. Allerdings breitet sich die Wellenfunktion deutlich schneller nach rechts aus. Dies bedeutet, daß der höher angeregte Zustand schneller dissoziiert, was auch theoretisch zu erwarten war (vgl. hierzu [43, 44]).

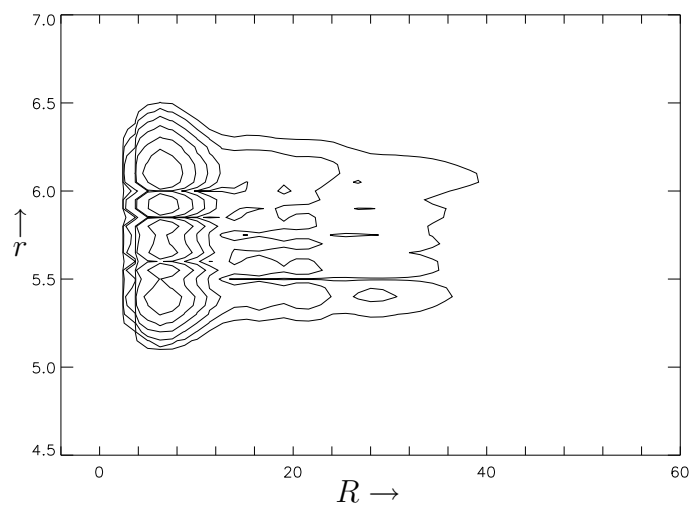
Ursprünglich war im Rahmen dieser Arbeit geplant, Berechnungen für den Zustand $v_r = 20$ mit aufzunehmen, um die daraus resultierenden Ergebnisse mit denen aus [41] zu vergleichen. Wie bereits erwähnt, wurde dort auf einem 256×256 -Gitter propagiert. Diese hohe Auflösung ist notwendig, um die Berechnungen mit ausreichender Genauigkeit durchführen zu können (vgl. hierzu die Anmerkungen zur Gitterauflösung am Ende von Abschnitt 5.3.4). Die in oben zitierter Arbeit beschriebenen Ergebnisse sind allerdings sehr knapp gehalten (was in einem per Internet veröffentlichten Artikel durchaus sinnvoll ist), insbesondere fehlen genauere Angaben über den verwendeten Darstellungsmodus (unklar ist z. B. die Zunahme der Wahrscheinlichkeitsdichte von einer Konturlinie zur nächsten), sodaß insbesondere ein grafischer Vergleich wenig

aussagekräftig wäre.

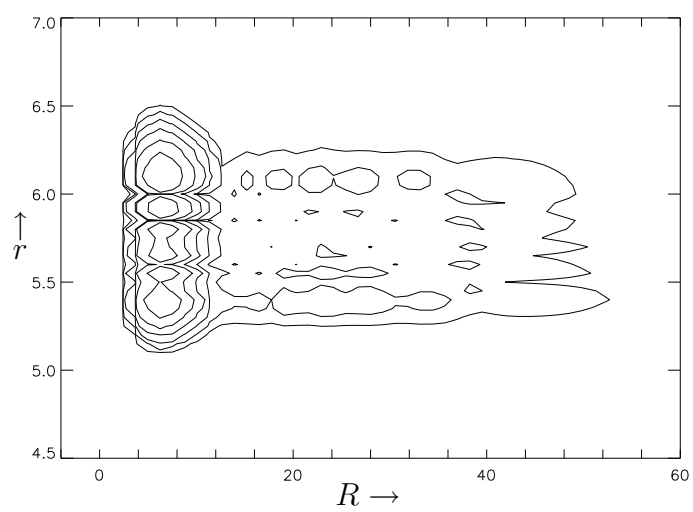
Wie bereits in Abschnitt 5.1.3 am eindimensionalen Jodidmolekül diskutiert, steigt der Rechenaufwand bei Erhöhung der Auflösung stark an. Hier im zweidimensionalen Fall steigt die Vektordimension durch den Übergang von 100 auf 256 Punkte auf das 2.56^2 -fache; die Hamiltonmatrix wird daher $2.56^4 \approx 43$ mal so groß. Daneben ist ein Faktor in der Größenordnung 3-4 durch die erhöhte obere Grenze des Energiebereichs zu erwarten. Der gesamte Rechenaufwand steigt deshalb auf das 130-170fache, weshalb von einer Durchführung dieser Berechnungen — insbesondere im Hinblick auf den Zeitrahmen dieser Arbeit — abgesehen wurde.



$t = 1.0$ ps

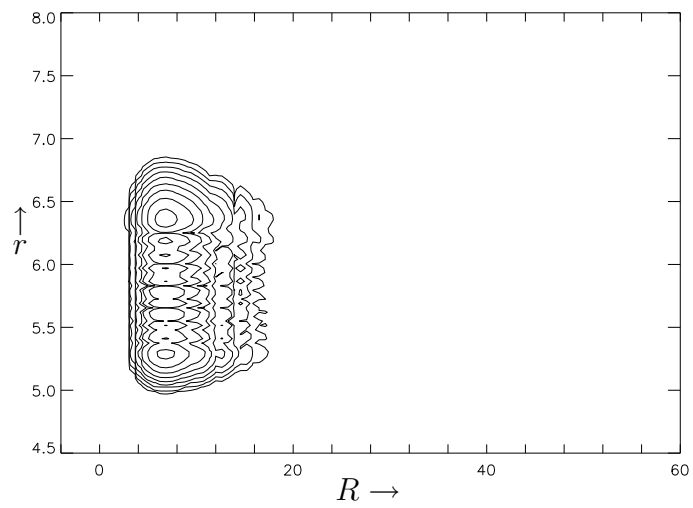


$t = 2.0$ ps

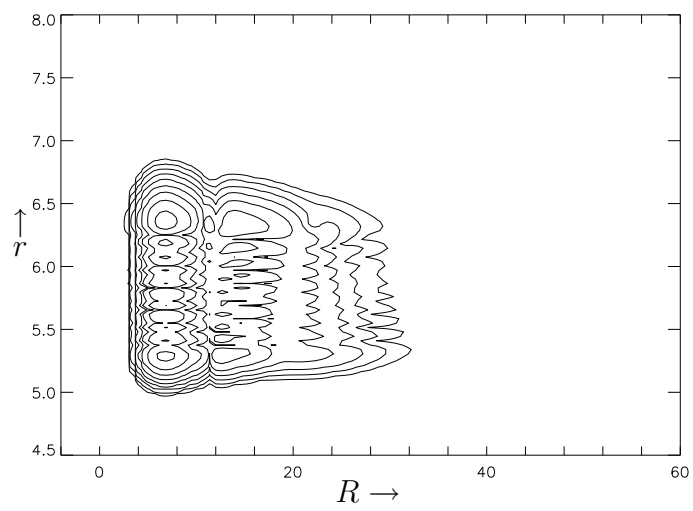


$t = 3.0$ ps

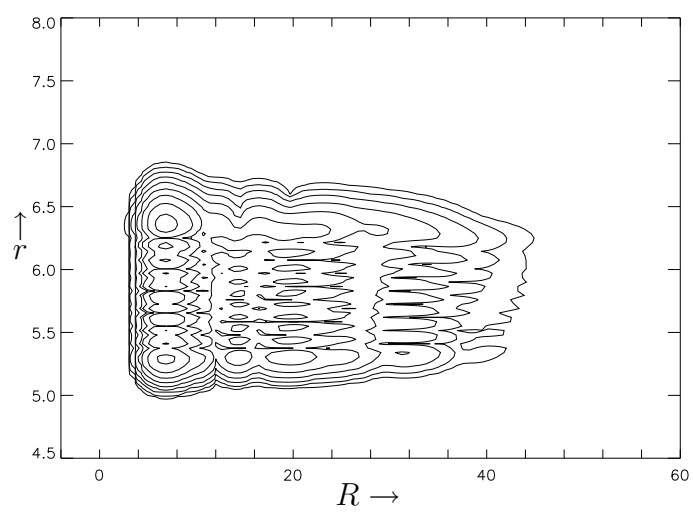
Abbildung 20: Propagation der Wellenfunktion mit $v_r = 5$ zu verschiedenen Zeiten t .



$t = 0.5$ ps



$t = 1.0$ ps



$t = 1.5$ ps

Abbildung 21: Propagation der Wellenfunktion mit $v_r = 10$ zu verschiedenen Zeiten t .

6 Zusammenfassung

In der vorliegenden Arbeit wurde versucht, einen Überblick über mögliche Verfahren zur Lösung der zeitabhängigen Schrödinger-Gleichung zu geben. Besondere Aufmerksamkeit wurde dabei auf die Extrapolationsmethoden gerichtet, da entsprechende Voruntersuchungen [15, 22] darauf schließen ließen, daß derartige Verfahren auch hier mit Erfolg eingesetzt werden könnten.

Diese Vermutung hat sich zum großen Teil bestätigt. Insbesondere bei der Berechnung der Matrixexponentialfunktion bieten sich durch den Einsatz von Beschleunigungsverfahren leistungsfähige Algorithmen zur numerischen Berechnung des Zeitentwicklungsoperators, die mit gängigen Verfahren durchaus konkurrieren können. Besonders der vektorielle ϵ -Algorithmus und die vektorielle \mathcal{J} -Transformation weisen hier gute Resultate auf.

Allerdings ist die Darstellung dieses Operators in Matrixform derzeit in konkreten physikalischen Problemen (insbesondere mit mehreren Freiheitsgraden) noch zu speicherintensiv, um auf Computern sinnvoll eingesetzt werden zu können. Da sich dies bei der raschen Entwicklung auf dem Computersektor sicher ändern wird, steht einem Einsatz der in Abschnitt 2 vorgestellten Matrixverfahren grundsätzlich nichts im Wege. Diese Verfahren bieten nämlich den Vorteil, daß z. B. verschiedene Ausgangswellenfunktionen, die denselben Hamiltonoperator besitzen und im selben Koordinatenbereich definiert sind, durch einfache Matrixmultiplikation mit dem zuvor berechneten Zeitentwicklungsoperator zu propagieren. Mehrfaches Hintereinanderschalten dieser Operation erlaubt es, die zeitliche Entwicklung eines Systems zu beobachten. Hier muß die Zeitentwicklungsmatrix nur einmal berechnet und abgespeichert werden und steht dann für alle folgenden Berechnungen zur Verfügung.

Scheidet die Verwendung von Matrizen aus, so bietet sich die Vektorpropagation an (Abschnitt 3). Auch hier ist der Einsatz von Extrapolationsmethoden möglich, wobei hier — anders als im Matrizenfall — kein allgemeingültiges „Rezept“ vorgestellt werden kann, wie die Berechnungen optimal durchgeführt werden können. Vielmehr spielen hier viele Faktoren eine Rolle, z. B. die Norm der Hamiltonmatrix, die Verteilung der Energieeigenwerte, der Skalierungsgrad, der Speicherbedarf usw., sodaß hier Vorversuche notwendig werden, um die Berechnungen zu optimieren. Hier zeigte es sich, daß die Beschleunigungsverfahren nicht mit den momentan üblicherweise verwendeten Verfahren konkurrieren können, was Rechenaufwand und Speicherbedarf betrifft. Zieht man allerdings die erzielbare Genauigkeit hinzu, so schneiden die Beschleunigungsalgorithmen deutlich besser ab und übertreffen zum Teil die „Standardverfahren“.

Somit erweisen sich die Extrapolationsverfahren als vielversprechende Kandidaten für weiterführende Untersuchungen auf diesem Gebiet. Mögliche Zielrichtungen wären z. B.

- die Abhängigkeit des Beschleunigungsverhaltens von der Struktur des Spektrums; einige Aussagen darüber sind bereits in Abschnitt 4.1.5 zu finden,

- sowie die Anwendung von Beschleunigungsverfahren auf die Folge der Tschebyscheff-Polynome; hier wären Methoden zur Konvergenzbeschleunigung von Entwicklungen nach Orthogonalpolynomen notwendig [22, Kap. 5.5], die in dieser Arbeit nicht behandelt wurden.

A Algorithmen

Im folgenden werden die verwendeten Beschleunigungsalgorithmen vorgestellt, wie sie mit MAPLE V Release 5.1 [39] verwendet wurden. Die D -dimensionalen Vektoren wurden dabei als $(D \times 1)$ -Matrizen übergeben, sodaß das Skalarprodukt $\vec{a} \cdot \vec{b}$ als $\text{tr}(\mathbf{a}^T \cdot \mathbf{b})$ formuliert wurde. Das hat den Vorteil, daß die Routinen auch direkt für Matrizen verwendet werden können. Die durchgeführten Tests lieferten durchweg ähnliche Resultate unabhängig davon, ob die Matrix in einen Vektor umgewandelt oder direkt als Argument übergeben wurde (für den weiter unten beschriebenen Matrix- \mathcal{J} -Algorithmus müssen die Argumente quadratische Matrizen sein). Deswegen wurde im folgenden auf eine explizite vektorielle Schreibweise der auftretenden Größen verzichtet.

A.1 Vektorieller ϵ -Algorithmus

Der vektorielle ϵ -Algorithmus wurde bereits in Kap. 2.4.1, Gl. (115) vorgestellt. Die in dieser Arbeit verwendete Implementierung dieses Algorithmus geht auf [63] zurück und berechnet beim n ten Aufruf die Gegendiagonale $\epsilon_0^{(n)}, \epsilon_1^{(n-1)}, \dots, \epsilon_n^{(0)}$, die in den Variablen `epsilon[n]`, `epsilon[n-1]`, `...`, `epsilon[0]` abgelegt wird. Abb. 22 veranschaulicht die Position der Rechengrößen in der ϵ -Tabelle.

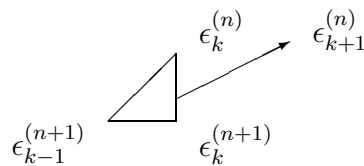


Abbildung 22: Ausschnitt aus der ϵ -Tabelle bei der Berechnung des vektoriellen ϵ -Algorithmus

Der Routine `epsvec` wird das aktuelle Folgeelement s_n in der Variablen `sofn` übergeben. Zur Berechnung werden zwei Hilfsvariable `aux1` und `aux2` benötigt. In `aux2` wird zuerst $\epsilon_{-1}^{(n+1)} = 0$ abgelegt, dann jeweils der Wert der Gegendiagonalen, der infolge der Berechnung des neuen Wertes überschrieben wird. Als angenommener Grenzwert wird `epsilon[0]` oder `epsilon[1]` in der Variablen `predic` zurückgegeben, je nachdem, ob n gerade oder ungerade ist, da die $\epsilon_k^{(n)}$ mit ungeradem k nur Hilfsgrößen sind. Die Verwendung der Variablen `tiny` verhindert, daß das Skalarprodukt eine vorgegebene Schranke unterschreitet, was andernfalls eine Division durch Null und einen damit verbundenen Programmabbruch zur Folge hätte.

```
epsvec:=proc(n,sofn,epsilon,predic)
local aux1,aux2,diff,j,tr,tiny;
tiny:=1e-60;
epsilon[n]:=evalm(sofn);
if n=0 then
```

```

    predic:=evalm(sofn);
else
    aux2:=evalm(0);
    for j from n to 1 by -1 do
        aux1:=evalm(aux2);
        aux2:=evalm(epsilon[j-1]);
        diff:=evalm(epsilon[j]-aux2);
        tr:=trace(htranspose(diff)*diff);
        if tr<tiny then tr:=tiny fi:
        epsilon[j-1]:=evalm(aux1+diff/tr);
    od;
    if (n mod 2)=0 then
        predic:=evalm(epsilon[0]):
    else
        predic:=evalm(epsilon[1]):
    fi:
fi:
end:

```

A.2 Topologische ϵ -Algorithmen

Von diesem Algorithmus existieren zwei Varianten, die hier im einzelnen vorgestellt werden sollen.

Der *erste* topologische ϵ -Algorithmus basiert auf der Rekursion (116) aus Kap. 2.4.1: Auch hier sind die zur Berechnung benötigten Ausschnitte der ϵ -Tabelle in Abb. 23 dargestellt.

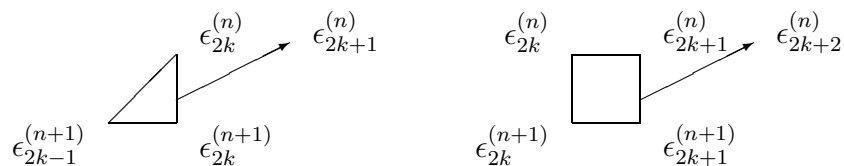


Abbildung 23: Ausschnitte aus der ϵ -Tabelle bei der Berechnung des ersten topologischen ϵ -Algorithmus

Bei der Implementierung dieses Algorithmus wurde versucht, ein Verfahren zu entwickeln, daß von der Konzeption dem des vektoriiellen ϵ -Algorithmus entspricht. Anders als dort ist hier zu unterscheiden, ob $\epsilon_k^{(n)}$ mit geradem oder ungeradem k berechnet werden, da für beide Fälle unterschiedliche Rekursionen zugrundeliegen. Insbesondere sind bei der Rekursionsvorschrift für gerade k Werte aus der vorletzten Gegendiagonalen nötig, die in einem zusätzlichen Feld `aux[]` abgelegt werden. Außerdem muß der Hilfsvektor y in der Variablen `y` an die Routine übergeben werden.

```
epstop1:=proc(n,sofn,y,epsilon,aux,predic)
```

```

local aux1,aux2,diff1,diff2,j,tr,tiny;
tiny:=1e-60;
epsilon[n]:=evalm(sofn);
aux[n]:=evalm(sofn);
if n=0 then
  predic:=evalm(sofn);
else
  aux2:=evalm(0);
  for j from n to 1 by -1 do
    aux1:=evalm(aux2);
    aux2:=evalm(epsilon[j-1]);
    diff1:=evalm(epsilon[j]-aux2);
    if (n-j) mod 2=0 then
      tr:=trace(transpose(y)&*diff1);
      if abs(tr)<tiny then tr:=tiny fi:
      epsilon[j-1]:=evalm(aux1+y/tr);
    else
      diff2:=evalm(aux1-aux[j-1]);
      tr:=trace(transpose(diff1)&*diff2);
      if abs(tr)<tiny then tr:=tiny fi:
      epsilon[j-1]:=evalm(aux1+diff2/tr);
      aux[j-1]:=evalm(epsilon[j-1]);
    fi;
  od;
  if (n mod 2)=0 then
    predic:=evalm(epsilon[0]);
  else
    predic:=evalm(epsilon[1]);
  fi;
fi:
end:

```

Für den *zweiten* topologischen ϵ -Algorithmus nach Gl. (117) ist die Rekursion für die $\epsilon_k^{(n)}$ mit geradem k verändert. Die Darstellung in der ϵ -Tabelle erfolgt in Abb. 24. Durch die Verschiebung der oberen Indizes in Zähler und Nenner ist das `aux[]`-Feld nicht mehr notwendig.

```

epstop2:=proc(n,sofn,y,epsilon,predic)
local aux1,aux2,diff1,diff2,j,tr,tiny;
tiny:=1e-60;
epsilon[n]:=evalm(sofn);
if n=0 then
  predic:=evalm(sofn);
else

```

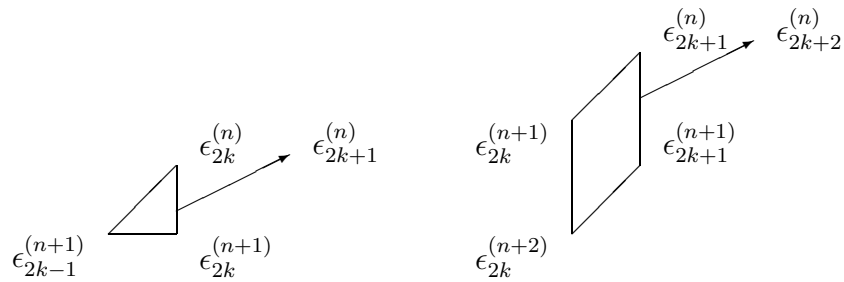


Abbildung 24: Ausschnitte aus der ϵ -Tabelle bei der Berechnung des zweiten topologischen ϵ -Algorithmus

```

aux2:=evalm(0);
for j from n to 1 by -1 do
  aux1:=evalm(aux2);
  aux2:=evalm(epsilon[j-1]);
  diff1:=evalm(epsilon[j]-aux2);
  if (n-j) mod 2=0 then
    tr:=trace(transpose(y)&*diff1);
    if abs(tr)<tiny then tr:=tiny fi;
    epsilon[j-1]:=evalm(aux1+y/tr);
  else
    diff2:=evalm(epsilon[j+1]-aux1);
    tr:=trace(transpose(diff1)&*diff2);
    if abs(tr)<tiny then tr:=tiny fi;
    epsilon[j-1]:=evalm(aux1+diff2/tr);
  fi;
od;
if (n mod 2)=0 then
  predic:=evalm(epsilon[0]);
else
  predic:=evalm(epsilon[1]);
fi;
fi;
end:

```

A.3 \mathcal{J} -Algorithmen

Die erste verwendete Variante \mathcal{J} -Algorithmus hat die in Kap. 2.4.3, Gl. 121 beschriebene Form.

Auch hier wurde — analog zu den weiter oben besprochenen ϵ -Algorithmen (man beachte die vertauschte Position von n und k !) — eine Implementierung gewählt, die die berechneten Werte $s_n^{(k)}$ bzw. $\omega_n^{(k)}$ in Form von Gegendiagonalen in den Feldern `ars []` bzw. `aro []` abspeichert. Diese Implementierung wurde von H. Homeier entwickelt. In

den Variablen `tr1`, `tr2` und `tr3` werden die Skalarprodukte des Nenners, des Zählers der s -Rekursion und des Zählers der ω -Rekursion gespeichert und anschließend die neuen Einträge berechnet. Mit `deltar` wird die Differenz $\Delta r_n^{(k)}$ der Hilfszahlenfolge $r_n^{(k)}$ an die Routine übergeben.

```
vj1:=proc(n,sofn,omeofn,ars,aro,deltar,predic)
local j,m,diff,tiny,tr1,tr2,tr3;
tiny:=1e-60;
ars[n]:=evalm(sofn);
aro[n]:=evalm(omeofn);
for j from n to 1 by -1 do
  m:=n-j+1;
  diff := htranspose(aro[j]-aro[j-1]);
  tr1 := trace(diff &* htranspose(diff));
  tr2 := trace(diff &* (ars[j]-ars[j-1]));
  tr3 := trace(diff &* aro[j-1]);
  if tr1<=tiny then tr1:=tiny fi;
  ars[j-1] := evalm(ars[j]-aro[j]*tr2/tr1);
  aro[j-1] := -evalm(aro[j]*tr3/tr1*deltar(j-1,m));
od;
predic:=evalm(ars[0]);
end:
```

Die andere eingesetzte Matrix-Variante ist etwas rechenaufwendiger. Sie beruht auf der allgemeinen Vektor- \mathcal{J} -Transformation nach Gl. (119). Für die Matrizenfolge $r_n^{(k)}$ wurde hier nicht $r_n^{(k)}1$ gewählt wie in obigem Algorithmus, sondern es wurde $r_n^{(k)} = r_n^{(k)}A$ gesetzt.

Die Implementierung erfolgte analog zum ersten Verfahren, nur wird statt der Differenz der Folgeelemente `deltar` die Matrix `A` und die skalare Hilfsfolge `r` übergeben. Außerdem sind die Argumente `sofn` und `omeofn` quadratische Matrizen und keine Vektoren.

```
mj:= proc(n,sofn,omeofn,ars,aro,r,A,predic)
local j,m,diff,tiny,tr1,tr2,tr3;
tiny:=1e-60;
ars[n]:=evalm(sofn);
aro[n]:=evalm(omeofn);
for j from n to 1 by -1 do
  m:=n-j;
  diff:=htranspose(aro[j]-aro[j-1]);
  tr1:=trace(diff &* htranspose(diff));
  tr2:=trace(diff &* (ars[j]-ars[j-1]));
  tr3:=trace(diff &* (A &* (aro[j]*r(j,m)-aro[j-1]*r(j-1,m)) ));
  if tr1<=tiny then tr1:=tiny fi;
  ars[j-1]:=evalm(ars[j]-aro[j]*tr2/tr1);
```

```

    aro[j-1]:=evalm(A &* aro[j]*r(j,m)-aro[j]*tr3/tr1);
od;
predic:=evalm(ars[0]);
end:

```

B Das DVR-Verfahren

B.1 Formaler Zugang

In diesem Abschnitt soll ein formaler Zugang zum DVR-Verfahren beschrieben werden, wie er auch unter [8, 9] zu finden ist.

Ausgangspunkt ist eine Menge von Basisfunktionen

$$\phi_\alpha(x), \quad j = 1, \dots, B \quad (243)$$

und eine Zahl von Gitterpunkten

$$x_\alpha, \quad \alpha = 1, \dots, G. \quad (244)$$

Es besteht keine Notwendigkeit, daß die Dimensionen der Basisfunktionen und der Punktbasis übereinstimmen müssen (vgl. dazu [9, S.1406]), deshalb wird im folgenden der allgemeinste Fall betrachtet.

Die Basis $\{\phi_j\}_B$ spannt einen B -dimensionalen Unterraum des Hilbertraums auf. Gesucht ist nun eine Darstellung, in der Integrale über die Basisvektoren durch (gewichtete) Summationen über die Gitterpunkte $\{x_\alpha\}_G$ approximiert werden.

Dazu setzt man

$$\psi_j(x) = \sum_{j=1}^B \phi_j(x) Q_{j\alpha} \quad (245)$$

mit einer $(B \times G)$ -Transformationsmatrix $Q = (Q_{j\alpha})$.

Nun gilt einerseits

$$\langle \psi_\alpha | \psi_\beta \rangle = \sum_{j=1}^B \sum_{k=1}^B Q_{j\alpha}^* Q_{k\beta} \langle \phi_j | \phi_k \rangle \quad (246)$$

$$= \sum_{j=1}^B \sum_{k=1}^B Q_{j\alpha}^* Q_{k\beta} S_{jk} \quad (247)$$

$$= (Q^\dagger S Q)_{\alpha\beta} \quad (248)$$

mit der Überlappmatrix

$$S = (S_{jk}) = \langle \phi_j | \phi_k \rangle; \quad (249)$$

andererseits soll ja

$$\langle \psi_\alpha | \psi_\beta \rangle \approx \sum_{\gamma=1}^G \omega_\gamma \psi_\alpha^*(x_\gamma) \psi_\beta(x_\gamma) \quad (250)$$

$$= \sum_{\gamma=1}^G \sum_{j=1}^B \sum_{k=1}^B \omega_\gamma \mathbf{Q}_{j\alpha}^* \mathbf{Q}_{k\beta} \phi_j^*(x_\gamma) \phi_k(x_\gamma) \quad (251)$$

$$\stackrel{!}{=} \delta_{\alpha\beta} \quad (252)$$

gelten.

Definiert man eine $(G \times B)$ -Matrix \mathbf{U} gemäß

$$\mathbf{U}_{\gamma j} = \sqrt{\omega_\gamma} \phi_j(x_\gamma), \quad (253)$$

so wird daraus

$$\delta_{\alpha\beta} \stackrel{!}{=} \sum_{\gamma=1}^G \sum_{j=1}^B \sum_{k=1}^B \mathbf{Q}_{j\alpha}^* \mathbf{U}_{\gamma j}^* \mathbf{U}_{\gamma k} \mathbf{Q}_{k\beta} \quad (254)$$

bzw.

$$\mathbf{1}_{G \times G} = \mathbf{Q}^\dagger \mathbf{U}^\dagger \mathbf{U} \mathbf{Q}. \quad (255)$$

Dies läßt sich erfüllen, wenn man \mathbf{Q} so wählt, daß

$$\mathbf{U} \mathbf{Q} = \mathbf{1}_{G \times G} \quad (256)$$

gilt. Für $G=B$ muß also

$$\mathbf{U} = \mathbf{Q}^{-1} \quad (257)$$

gelten.

Ist die gewählte Basis $\{\phi_j\}_B$ orthonormal bzgl. der Integration (vgl. Gl. (10) in Abschnitt 1.3.1), so ist

$$S_{jk} = \delta_{jk}. \quad (258)$$

Gln. (248) und (251) vereinfachen sich dann zu

$$\langle \psi_\alpha | \psi_\beta \rangle = (\mathbf{Q}^\dagger \mathbf{Q})_{\alpha\beta} \stackrel{!}{=} \delta_{\alpha\beta}, \quad (259)$$

sodaß auch

$$\mathbf{Q}^\dagger \mathbf{Q} = \mathbf{1}_{G \times G} \quad (260)$$

gilt. Dagegen ist unabhängig von der Wahl der Basis die Überlappmatrix \mathbf{S} näherungsweise durch

$$\mathbf{S} = \mathbf{U}^\dagger \mathbf{U} \quad (261)$$

gegeben.

Auch in dieser Darstellung ist die potentielle Energie diagonal, wie sich durch

$$\langle \psi_\alpha | \hat{V} | \psi_\beta \rangle \approx \sum_{\gamma=1}^G \sum_{j=1}^B \sum_{k=1}^B \omega_\gamma \phi_j^*(x_\gamma) \mathbf{Q}_{k\alpha}^* V(x_\gamma) \phi_k(x_\gamma) \mathbf{Q}_{k\beta} \quad (262)$$

$$= \sum_{\gamma=1}^G \sum_{j=1}^B \sum_{k=1}^B \mathbf{Q}_{j\alpha}^* \mathbf{U}_{\gamma j}^* V(x_\gamma) \mathbf{U}_{\gamma k} \mathbf{Q}_{k\beta} \quad (263)$$

$$= \sum_{\gamma=1}^G \delta_{\alpha\gamma} V(x_\gamma) \delta_{\gamma\beta} \quad (264)$$

$$= V(x_\alpha) \delta_{\alpha\beta} \quad (265)$$

leicht nachrechnen läßt.

B.2 Ein universelles DVR-Verfahren

In [6] wird ein DVR-Verfahren vorgestellt, das als „black box“-Verfahren konzipiert und universell einsetzbar ist. Nachdem dieses Verfahren auch in der vorliegenden Arbeit eingesetzt wurde, soll es hier genauer beschrieben werden.

Im vorliegenden Fall wird ein eindimensionales Problem vorausgesetzt und deshalb statt x die Ortskoordinate r gewählt. Außerdem werden ebensoviele Basisfunktionen wie Gitterpunkte eingesetzt, es gilt also

$$B = G = D$$

Der Hamiltonoperator des betrachteten Problems habe die Form

$$\hat{H} = \hat{T} + \hat{V}, \quad \text{wobei}$$

$$\hat{T} = -\frac{1}{2m} \frac{\partial^2}{\partial r^2} \quad (266)$$

$$\hat{V} = V(r). \quad (267)$$

Weiterhin werden D äquidistante Gitterpunkte r_α gewählt, die durch

$$\Delta r = \frac{r_{\max} - r_{\min}}{D + 1} \quad (268)$$

und

$$r_\alpha = r_{\min} + \alpha \Delta r, \quad \alpha = 1, 2, \dots, D \quad (269)$$

im Bereich $[r_{\min}, r_{\max}]$ festgelegt sind.

Als Basisfunktionen werden die D unteren (normierten) Eigenfunktionen des Kastenpotentials der Form

$$\phi_j(r_\alpha) = \langle r_\alpha | j \rangle = \sqrt{\frac{2}{D+1}} \sin\left(\frac{\alpha j \pi}{D+1}\right), \quad j = 1, 2, \dots, D \quad (270)$$

verwendet.

Als erstes werden Matrixelemente der Form $T_{\alpha\beta}$, ($\alpha \neq \beta$) berechnet. Es gilt:

$$\begin{aligned} T_{\alpha\beta} &= \langle r_\alpha | \hat{T} | r_\beta \rangle \\ &= \sum_{j=1}^D \langle r_\alpha | \hat{T} | j \rangle \langle j | r_\beta \rangle \\ &= -\frac{1}{2m} \frac{2}{D+1} \sum_{j=1}^D \sin\left(\frac{\alpha j \pi}{D+1}\right) \frac{\partial^2}{\partial r_\beta^2} \sin\left(\frac{\beta j \pi}{D+1}\right). \end{aligned}$$

Aus

$$\sin\left(\frac{\beta j \pi}{D+1}\right) = \sin\left(\frac{(r_\beta - r_{\min})j\pi}{r_{\max} - r_{\min}}\right) \quad (271)$$

folgt

$$\frac{\partial^2}{\partial r_\beta^2} \sin\left(\frac{\beta j \pi}{D+1}\right) = -j^2 \frac{\pi^2}{(r_{\max} - r_{\min})^2} \sin\left(\frac{\beta j \pi}{D+1}\right); \quad (272)$$

somit ergibt sich

$$\begin{aligned} T_{\alpha\beta} &= +\frac{1}{2m} \frac{2}{D+1} \frac{\pi^2}{(r_{\max} - r_{\min})^2} \sum_{j=1}^D j^2 \sin\left(\frac{\alpha j \pi}{D+1}\right) \sin\left(\frac{\beta j \pi}{D+1}\right) \\ &= \frac{1}{2m} \frac{2}{D+1} \frac{\pi^2}{(r_{\max} - r_{\min})^2} \frac{1}{2} \sum_{j=1}^D j^2 \left[\cos\left(\frac{(\alpha - \beta)j\pi}{D+1}\right) - \cos\left(\frac{(\alpha + \beta)j\pi}{D+1}\right) \right] \end{aligned} \quad (273)$$

Nun substituiert man

$$C^\pm := \frac{(\alpha \pm \beta)\pi}{D+1} \quad (274)$$

und nutzt die Beziehung

$$\sum_{j=1}^D j^2 \cos(jC^\pm) = -\frac{\partial^2}{\partial C^{\pm 2}} \operatorname{Re} \sum_{j=1}^D e^{ijC^\pm}. \quad (275)$$

Letztere Summe läßt sich über die geometrische Reihe geschlossen darstellen; es gilt

$$\sum_{j=1}^D e^{ijC^\pm} = e^{iC^\pm} \frac{e^{iDC^\pm} - 1}{e^{iC^\pm} - 1}. \quad (276)$$

Erweitert man Zähler und Nenner mit $-i(e^{-iC^\pm} + 1)$, so wird der Nenner reell und der Realteil der Summe läßt sich leicht berechnen. Nach einigen trigonometrischen Umformungen erhält man so

$$\operatorname{Re} \sum_{j=1}^D e^{ijC^\pm} = -\frac{1}{2} + \frac{1}{2} \frac{\sin(D+1/2)C^\pm}{\sin C^\pm/2}. \quad (277)$$

Zur Berechnung der zweiten Ableitung wird die Quotientenregel herangezogen und zur weiteren Umformung und Vereinfachung folgende Beziehungen verwendet:

$$\sin(D + 1/2)C^\pm = \sin(D + 1)C^\pm \cos C^\pm/2 - \cos(D + 1)C^\pm \sin C^\pm/2 \quad (278)$$

$$\cos(D + 1/2)C^\pm = \cos(D + 1)C^\pm \cos C^\pm/2 + \sin(D + 1)C^\pm \sin C^\pm/2 \quad (279)$$

$$\sin(D + 1)C^\pm = \sin(\alpha \pm \beta)\pi = 0 \quad (280)$$

$$\cos(D + 1)C^\pm = \cos(\alpha \pm \beta)\pi = (-1)^{\alpha \pm \beta} \quad (281)$$

$$(-1)^{\alpha + \beta} = (-1)^{\alpha - \beta}. \quad (282)$$

Daraus folgt schließlich

$$\frac{\partial^2}{\partial C^{\pm 2}} \operatorname{Re} \sum_{j=1}^D e^{ijC^\pm} = (-1)^{\alpha - \beta} (D + 1) \left[(D + 1) - \frac{1}{\sin^2 C^\pm/2} \right]. \quad (283)$$

Eingesetzt in (273) erhält man das Resultat

$$\mathsf{T}_{\alpha\beta} = \frac{\pi^2}{2m} \frac{(-1)^{\alpha - \beta}}{2(r_{\max} - r_{\min})^2} \left[\frac{1}{\sin^2 \left(\frac{(\alpha - \beta)\pi}{2(D + 1)} \right)} - \frac{1}{\sin^2 \left(\frac{(\alpha + \beta)\pi}{2(D + 1)} \right)} \right], \quad (\alpha \neq \beta). \quad (284)$$

Ist dagegen $\alpha = \beta$, so wird $C^- = 0$ und somit $\cos nC^- = 1$; setzt man dies in (273) ein, folgt

$$\mathsf{T}_{\alpha\alpha} = \frac{1}{2m} \frac{2}{D + 1} \frac{\pi^2}{(r_{\max} - r_{\min})^2} \frac{1}{2} \sum_{j=1}^D j^2 [1 - \cos jC^+]. \quad (285)$$

Die Summation über den ersten Summanden in der eckigen Klammer liefert

$$\sum_{j=1}^D j^2 = \frac{D(D + 1)(2D + 1)}{6}, \quad (286)$$

der zweite Summand wird mit $(-1)^{\alpha + \alpha} = (-1)^{2\alpha} = 1$ zu

$$\sum_{j=1}^D j^2 \cos jC^+ = (D + 1) \left[(D + 1) - \frac{1}{\sin^2 C^+/2} \right]; \quad (287)$$

für die Diagonalelemente $\mathsf{T}_{\alpha\alpha}$ ergibt sich also

$$\mathsf{T}_{\alpha\alpha} = \frac{\pi^2}{4m(r_{\max} - r_{\min})^2} \left[\frac{2(D + 1)^2 + 1}{3} - \frac{1}{\sin^2 \left(\frac{\alpha\pi}{D + 1} \right)} \right]. \quad (288)$$

Für die potentielle Energie gilt:

$$\begin{aligned}
V_{\alpha\beta} &= \langle r_\alpha | \hat{V} | r_\beta \rangle \\
&= \sum_{j=1}^D \langle r_\alpha | \hat{V} | j \rangle \langle j | r_\beta \rangle \\
&= \frac{2}{D+1} \sum_{j=1}^D \sin\left(\frac{\alpha j \pi}{D+1}\right) V(r_\alpha) \sin\left(\frac{\alpha j \pi}{D+1}\right) \\
&= V(r_\alpha) \frac{2}{D+1} \sum_{j=1}^D \sin\left(\frac{\alpha j \pi}{D+1}\right) \sin\left(\frac{\beta j \pi}{D+1}\right) \\
&= V(r_\alpha) \frac{2}{D+1} \frac{1}{2} \sum_{j=1}^D [\cos jC^- - \cos jC^+].
\end{aligned} \tag{289}$$

Die weiteren Rechenschritte laufen dann analog ab; es kann Gl. (277) verwendet werden. Aus ihr folgt

$$\sum_{j=1}^D [\cos jC^- - \cos jC^+] = \delta_{\alpha\beta}, \tag{290}$$

insgesamt also

$$V_{\alpha\beta} = V(r_\alpha) \delta_{\alpha\beta}. \tag{291}$$

Das Potential ist also diagonal, was theoretisch bereits an anderer Stelle gezeigt wurde (Gl. (262) in B.1).

Setzt man insbesondere $V(r_\alpha) \equiv 1$ für alle r_α , so folgt aus (289) und (291) die diskrete Orthogonalitätsrelation für die DVR-Darstellung

$$\langle r_\alpha | r_\beta \rangle = \delta_{\alpha\beta}. \tag{292}$$

Abschließend hier noch einmal die Ergebnisse dieses Abschnitts [6]:

$$T_{\alpha\beta} = \frac{\pi^2}{4m} \frac{(-1)^{\alpha-\beta}}{(r_{\max} - r_{\min})^2} \times \begin{cases} \frac{1}{\sin^2\left(\frac{(\alpha-\beta)\pi}{2(D+1)}\right)} - \frac{1}{\sin^2\left(\frac{(\alpha+\beta)\pi}{2(D+1)}\right)}, & \alpha \neq \beta \\ \frac{2(D+1)^2 + 1}{3} - \frac{1}{\sin^2\left(\frac{\alpha\pi}{D+1}\right)}, & \alpha = \beta \end{cases} \tag{293}$$

$$V_{\alpha\beta} = V(r_\alpha) \delta_{\alpha\beta}. \tag{294}$$

C Optische Potentiale

Um störende Randeinflüsse wie z. B. Reflexionen an den Rändern auf endlichen Gittern zu minimieren, werden in der Praxis häufig negative imaginäre Potentiale (NIPs) —

auch *optische Potentiale* genannt — eingesetzt. Wie die erste Bezeichnung nahelegt, sind dies Potentiale der Form

$$V_{\text{opt}}(r) = -i f(r), \quad \text{wobei} \quad f(r) \geq 0. \quad (295)$$

Sehr häufig (siehe z. B. [7, 64]) werden lineare Potentialformen gewählt, also Potentiale der Form

$$V_{\text{opt}}(r) = \begin{cases} -i V_{\text{opt}}^0 \frac{r - r_i}{r_f - r_i}, & \text{für} \quad r_i \leq r \leq r_f, \\ 0 & \text{sonst.} \end{cases} \quad (296)$$

Dabei muß sowohl die Steigung V_{opt}^0 als auch der Bereich $\Delta r = r_i - r_f$ bestimmte Bedingungen erfüllen, damit die Summe aus Reflexion R und Transmission T minimal wird. Genauere Untersuchungen hierzu finden sich in [64]. Die lineare Form ist aber nicht die einzig mögliche; in [65] wurden 5 verschiedene Potentialformen (linear, quadratisch, kubisch, quartisch und exponentiell) untersucht und tabelliert.

Zur Anpassung des optischen Potentials an die zu absorbierende Wellenfunktion muß dessen Energie E bekannt sein. Eine weitere Bestimmungsgröße ist die charakteristische Wellenlänge λ . Diese berechnet sich aus E und der Masse m gemäß

$$\lambda = \frac{h}{\sqrt{2mE}}. \quad (297)$$

Durch die Einführung eines solchen Potentials wird der Hamiltonoperator nicht-hermitesch, somit ist der Zeitentwicklungsoperator nicht mehr unitär. Das bedeutet, daß die Tschebyscheff-Entwicklung von \hat{U} nicht mehr ohne weiteres möglich ist, da beispielsweise der Bereich der Eigenwerte von der reellen Achse auf die komplexe Zahlenebene verschoben wird. In [65] wird deshalb nur der hermitesche Anteil des Hamiltonoperators zur Bestimmung des Energiebereichs herangezogen, bei der Reskalierung nach (72) jedoch der volle Hamiltonoperator.

In der vorliegenden Arbeit wurde ein anderer Weg beschritten, der auf eigenen Überlegungen beruht: Sei \hat{H} der vollständige Hamiltonoperator, definiert durch

$$\hat{H} = \hat{H}_0 + \hat{V}_{\text{opt}}, \quad (298)$$

wobei \hat{H}_0 die physikalische Situation ohne Dämpfung beschreibt. Dann läßt sich der Zeitentwicklungsoperator in Anlehnung an (142) in 3.4 approximieren durch

$$e^{-i\hat{H}t} \approx e^{-i(\hat{V}_{\text{opt}}/2)t} e^{-i\hat{H}_0 t} e^{-i(\hat{V}_{\text{opt}}/2)t}. \quad (299)$$

Wir definieren nun den Dämpfoperator \hat{D} :

$$\hat{D} = e^{-i(\hat{V}_{\text{opt}}/2)t}. \quad (300)$$

Somit gilt näherungsweise

$$e^{-i\hat{H}t} = \hat{D} e^{-i\hat{H}_0 t} \hat{D}. \quad (301)$$

Der Fehler liegt hier wieder in der Größenordnung $O(t^3)$.

Setzt man nun in (301) die Tschebyscheff-Entwicklung (79) ein, so ergibt sich

$$\begin{aligned} e^{-i\hat{H}t} &= \hat{\mathfrak{D}} e^{-i\hat{H}_0 t} \hat{\mathfrak{D}} \\ &\approx \hat{\mathfrak{D}} \left(\sum_{n=0}^N a_n T_n \right) \hat{\mathfrak{D}} \\ &= \sum_{n=0}^N a_n \hat{\mathfrak{D}} T_n \hat{\mathfrak{D}} \\ &= \sum_{n=0}^N a_n \tilde{T}_n, \end{aligned}$$

wenn man die „gedämpften“ Tschebyscheff-Polynome

$$\tilde{T}_n = \hat{\mathfrak{D}} T_n \hat{\mathfrak{D}} \quad (302)$$

einführt.

Wird nun (302) in die Rekursion der Tschebyscheff-Polynome nach (63) eingesetzt, so ergibt sich für die ersten beiden Polynome (\hat{H}_0 wird der Einfachheit halber als auf 1 normiert betrachtet)

$$\tilde{T}_0 = \hat{\mathfrak{D}} T_0 \hat{\mathfrak{D}} = \hat{\mathfrak{D}}^2, \quad \tilde{T}_1 = \hat{\mathfrak{D}} \hat{H}_0 \hat{\mathfrak{D}},$$

während für die eigentliche Rekursion

$$\begin{aligned} \tilde{T}_{n+1} &= \hat{\mathfrak{D}} T_{n+1} \hat{\mathfrak{D}} \\ &= \hat{\mathfrak{D}} (2\hat{H}_0 T_n - T_{n-1}) \hat{\mathfrak{D}} \\ &= 2\hat{\mathfrak{D}} \hat{H}_0 T_n \hat{\mathfrak{D}} - \hat{\mathfrak{D}} T_{n-1} \hat{\mathfrak{D}} \end{aligned}$$

gilt. Der zweite Summand ist nach Definition (302) gleich \tilde{T}_{n-1} ; beim ersten Term läßt sich wieder näherungsweise \hat{H}_0 mit dem führenden $\hat{\mathfrak{D}}$ vertauschen. Damit folgt

$$\tilde{T}_{n+1} = 2\hat{H}_0 \tilde{T}_n - \tilde{T}_{n-1}; \quad (303)$$

somit gilt die Rekursionsbeziehung (63) bis auf einen Fehler der Ordnung t^3 auch für die gedämpften Tschebyscheff-Polynome. Mit diesen so definierten Polynomen läßt sich natürlich auch wieder eine vektorielle Rekursion definieren mit der Ersetzung

$$\psi_n \longrightarrow \tilde{\psi}_n = \tilde{T}_n \psi(0) \quad (304)$$

in (137).

Auf der Grundlage der Vektorpropagation mit gedämpften Polynomen wurde ein Programm realisiert, daß an einem einfachen Beispiel den Einfluß eines optischen Potentials zeigt. Im folgenden wird zunächst ohne optisches Potential propagiert.

Der Hamiltonoperator des betrachteten Modellsystems lautet

$$\hat{H}_0 = \hat{T} + \hat{V}, \quad (305)$$

wobei $\hat{V} = V(r)$ ein Gauss-förmiges abstoßendes Potential der Form

$$V(r) = De^{-\alpha r^2}$$

mit den Parametern $D = 1000 \text{ cm}^{-1}$, $\alpha = 0.2 \text{ a}_0^{-1}$ darstellt; an der Stelle $r_{loc} = 10.0 \text{ a}_0$ wurde ein ebenfalls Gauss-förmiges Wellenpaket mit der Masse $m = 1.0 \text{ amu}$ und $\beta = 0.8 \text{ a}_0^{-1}$, beschrieben durch

$$\psi_0(r) \propto e^{-\beta(r-r_{loc})^2}$$

angenommen. Abbildung 25 zeigt die Ausgangssituation. Nach oben ist die Energie in cm^{-1} angegeben, nach rechts die Anzahl der Gitterpunkte. Propagiert wurde im Bereich von $r_{\min} = 0$ bis $r_{\max} = 50.0 \text{ a}_0$, unterteilt in $D = 256$ Punkte. Als Zeitschritt wurde $T = 0.1 \text{ ps}$ gewählt. In allen folgenden Abbildungen ist der Einfachheit halber als Abszisse der Laufindex α statt der Ortskoordinate r_α angetragen.

Die Wellenfunktion ist hier nicht normiert und wurde stark vergrößert gezeichnet, um das Auseinanderlaufen des Wellenpakets gut verfolgen zu können (die Größenverhältnisse der Wellenfunktionen zu den einzelnen Propagationsschritten sind zueinander maßstabsgetreu).

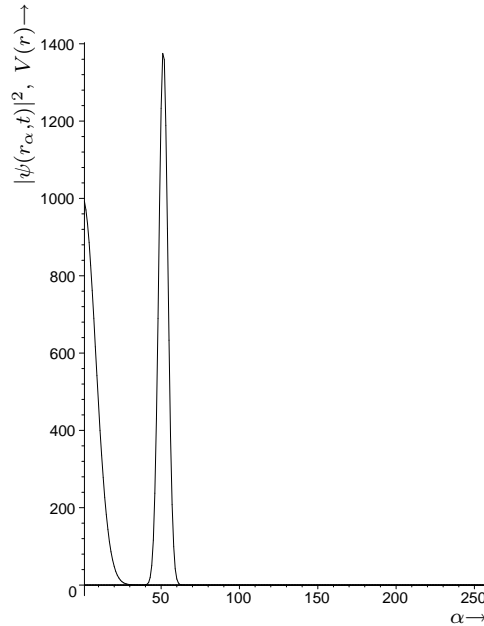


Abbildung 25: Betragsquadrat $|\psi_0(r_\alpha)|^2$ zum Zeitpunkt $t = 0$. Die Maßzahlen auf der linken Achse beziehen sich auf das Potential am linken Rand und sind in cm^{-1} angegeben.

In Abbildung 26 ist die Wellenfunktion zu späteren Zeitpunkten abgebildet. Man erkennt, wie das Wellenpaket erst auseinanderläuft und sich dann in mehrere Maxima mit nach außen hin abnehmender Intensität aufspaltet. Ab $t = 7T$ sieht man hochfrequente Überlagerungen der Wellenberge, die durch Reflexionen am Gitterrand hervorgerufen werden und sich in der Folgezeit nach links ausbreiten.

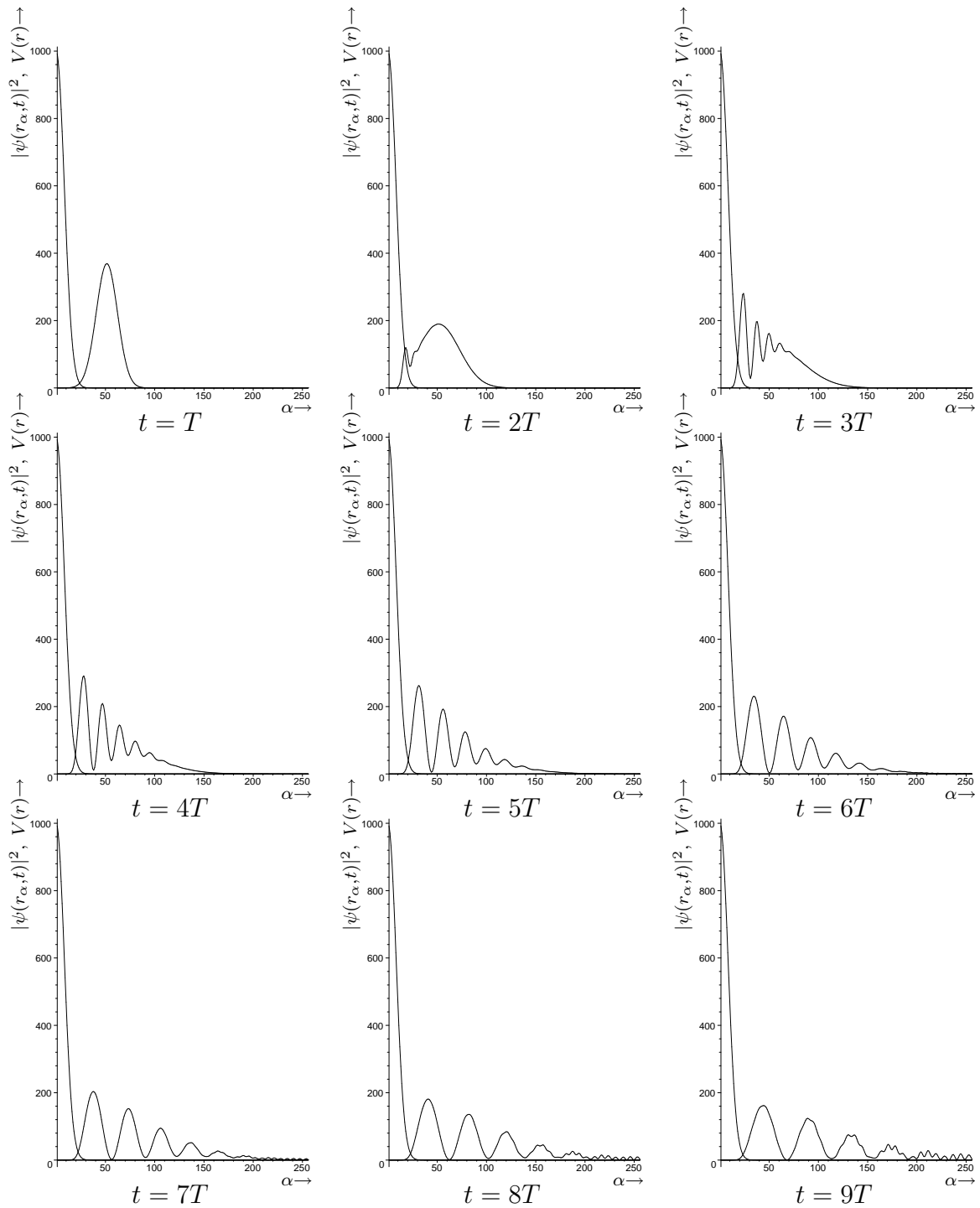


Abbildung 26: Betragsquadrat $|\psi(r_\alpha, t)|^2$ zu verschiedenen Zeiten t . Links ist das Modellpotential eingezeichnet. Ab $t = 7T$ werden hochfrequente Anteile am rechten Rand reflektiert und überlagern die Wellenfunktion.

In einem weiteren Durchlauf wurde nun ein optisches Potential eingeführt, um diese störenden Randeinflüsse zu minimieren. Das Programm berechnete die mittlere Ener-

gie²² des Wellenpakets mittels

$$E = \frac{\langle \psi_0 | \hat{H}_0 | \psi_0 \rangle}{\langle \psi_0 | \psi_0 \rangle} \quad (307)$$

zu $E = 17.6 \text{ cm}^{-1}$ und daraus die charakteristische Wellenlänge nach Gl. (297) zu $\lambda = 11.6 a_0$. Das Potential wurde am rechten Rand plaziert; somit ergibt sich das Potential nach Gl. (296), wenn man $V_{\text{opt}}^0 = 38.7 \text{ cm}^{-1}$, $r_\alpha = r_{\text{max}} - \lambda = 38.4 a_0$ und $r_f = r_{\text{max}} = 50.0 a_0$ setzt²³.

Die Grafiken in Abbildung 27 zeigen am rechten Rand das absorbierende Potential (hier reell gezeichnet) und die zeitliche Entwicklung des Wellenpaketes. Man erkennt, daß für die Zeit von $t = 0T$ bis $6T$ die Abbildungen mit denen aus Abb. 26 übereinstimmen, ab $t = 7T$ jedoch der hochfrequente Randeinfluß hier nicht erkennbar ist.

²²Für Wellenpakete mit breiter Energieverteilung wird in [65] eine empirische Formel zur Bestimmung der optimalen Energie angegeben; sie lautet:

$$E^* = \log^{-1} [0.6242 \log(E_{\text{min}}) + 0.3759 \log(E_{\text{max}})]. \quad (306)$$

Diese Formel ist für lineare und kubische Potentiale getestet worden.

Im vorliegenden Beispiel sollte der Einsatz von NIPs in erster Linie qualitativ gezeigt werden, deshalb wurde (306) hier der Einfachheit halber nicht angewandt.

²³Die benötigten Werte wurden aus den Tabellen in [65] entnommen; hierfür ergibt sich eine Dämpfung auf weniger als 0.3 % der ursprünglichen Wellenfunktion durch das optische Potential.

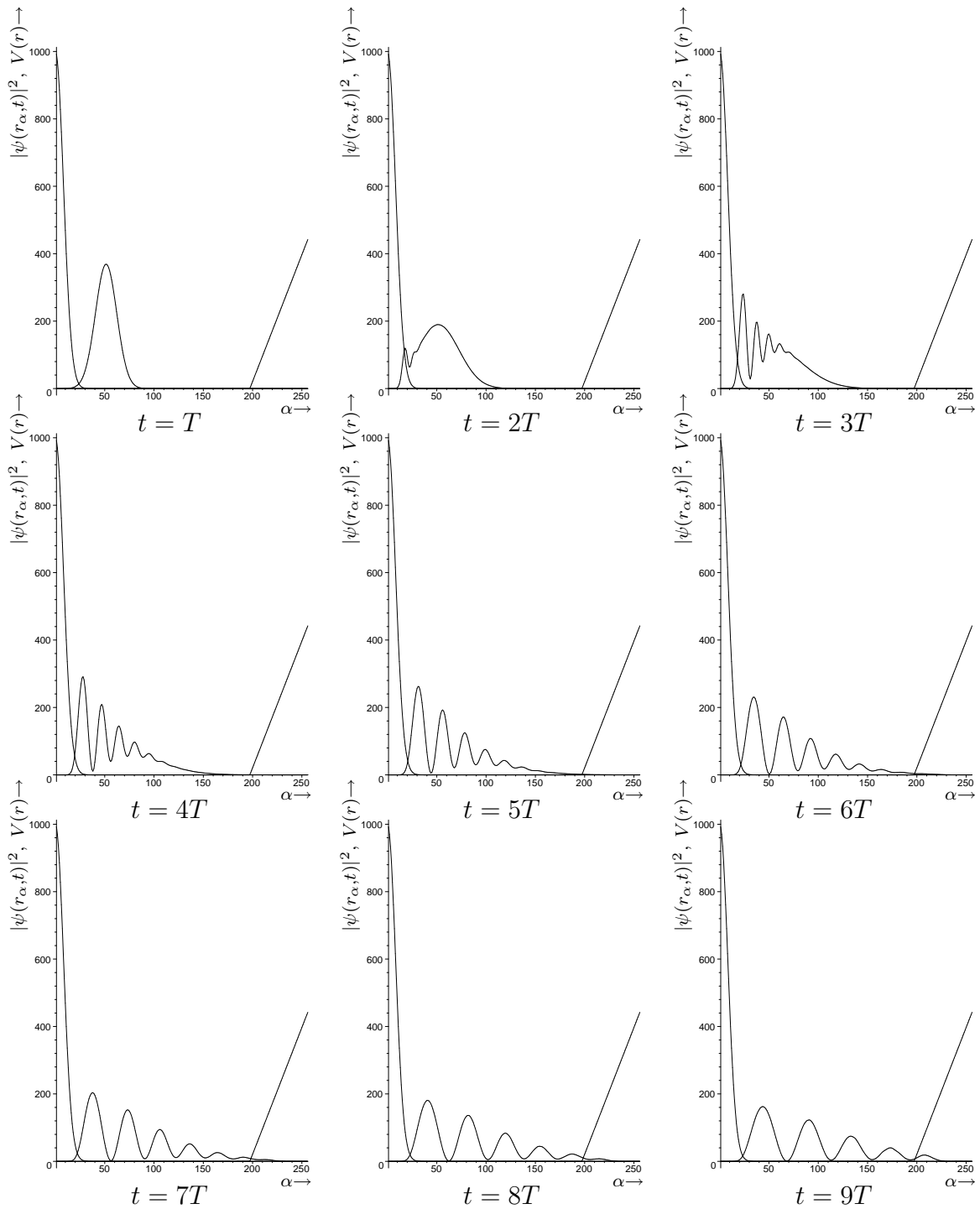


Abbildung 27: Betragsquadrat $|\psi(r_\alpha, t)|^2$ zu verschiedenen Zeiten t . Am rechten Rand ist der Verlauf des absorbierenden Potentials eingezeichnet. Hier sind im Gegensatz zu Abb. 26 keine Störungen vom rechten Rand her erkennbar.

Literatur

- [1] J. Schnakenberg. *Algorithmen in der Quantentheorie und Statistischen Physik*. Verlag Zimmermann-Neufang, Ulmen, 1995.
- [2] E. A. McCullough and R. E. Wyatt. Quantum dynamics of the collinear (H,H₂) reaction. *J. Chem. Phys.*, 51 (2):1253–1254, 1969.
- [3] E. A. McCullough and R. E. Wyatt. Dynamics of the collinear H+H₂ reaction: I. probability density and flux. *J. Chem. Phys.*, 54 (8):3578–3591, 1971.
- [4] E. A. McCullough and R. E. Wyatt. Dynamics of the collinear H+H₂ reaction: II. energy analysis. *J. Chem. Phys.*, 54 (8):3592–3600, 1971.
- [5] K. C. Kulander. Collision induced dissociation in collinear H+H₂: Quantum mechanical probabilities using the time-dependent wavepacket approach. *J. Chem. Phys.*, 69 (11):5064–5072, 1978.
- [6] D. T. Colbert and W. H. Miller. A novel discrete variable representation for quantum mechanical reactive scattering via the *S*-matrix Kohn method. *J. Chem. Phys.*, 96 (3):1982–1991, 1992.
- [7] N. Balakrishnan, C. Kalyanaraman, and N. Sathyamurthy. Time-dependent quantum mechanical approach to reactive scattering and related processes. *Physics report*, 280:97–144, 1997.
- [8] J. V. Lill, G. A. Parker, and J. C. Light. Discrete variable representations and sudden models in quantum scattering theory. *Chem. Phys. Lett.*, 89 (6):483–489, 1982.
- [9] J. C. Light, I. P. Hamilton, and J. V. Lill. Generalized discrete variable approximation in quantum mechanics. *J. Chem. Phys.*, 82 (3):1400–1409, 1985.
- [10] Z. Bačić and J. C. Light. Theoretical methods for rovibrational states of floppy molecules. *Annu. Rev. Phys. Chem.*, 40:469–498, 1989.
- [11] E. Oran Brigham. *FFT: schnelle Fourier-Transformation*. R. Oldenbourg Verlag GmbH, München, 1982.
- [12] I. A. Bronstein, K. A. Semendjajev, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch, Thun, Frankfurt am Main, zweite edition, 1995.
- [13] C. B. Moler and C. F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 20:801–836, 1978.
- [14] R. C. Ward. Numerical computation of the matrix exponential with accuracy estimate. *SIAM J. Numer. Anal.*, 14:600–614, 1977.
- [15] H. H. H. Homeier. On the computation of functions of matrices. In S. M. Bachrach, W. Hase, D. B. Boyd, H. S. Rzepa, and S. K. Gray, editors, *Proceedings of the First Electronic Computational Chemistry Conference*, CD-ROM (ARI-CD 101). ARInternet Corporation, 8201 Corporate Drive Suite 1100, Landover, MD 20785, U.S.A., 1994. Paper 53.
- [16] H. Tal-Ezer and R. Kosloff. An accurate and efficient scheme for propagating the time dependent Schrödinger equation. *J. Chem. Phys.*, 81:3967–3971, 1984.
- [17] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover Publications, Inc., New York, 1970.
- [18] Gene Golub and James M. Ortega. *Scientific Computing. An Introduction with parallel computing*. Academic Press, Inc., 1993.
- [19] R. Schäfer and R. Blendowske. On the numerical solution of the time-dependent Schrödinger equation. *J. Comput. Phys.*, 119:206–208, 1995.

- [20] G. A. Baker, Jr. and P. Graves-Morris. *Padé approximants*. Cambridge U.P., Cambridge (GB), second edition, 1996.
- [21] Yung-Ya Lin and Lian-Pin Hwang. Efficient computation of the matrix exponential using Padé approximation. *Computers Chem.*, 16:285–293, 1992.
- [22] H. H. H. Homeier. *Extrapolationsverfahren für Zahlen-, Vektor- und Matrizenfolgen und ihre Anwendung in der Theoretischen und Physikalischen Chemie*. Habilitation thesis, Universität Regensburg, 1996.
- [23] P. Wynn. On a device for computing the $e_m(S_n)$ transformation. *Math. Tables Aids Comput.*, 10:91–96, 1956.
- [24] D. Shanks. Non-linear transformations of divergent and slowly convergent sequences. *J. Math. and Phys. (Cambridge, Mass.)*, 34:1–42, 1955.
- [25] D. Levin. Development of non-linear transformations for improving convergence of sequences. *Int. J. Comput. Math. B*, 3:371–388, 1973.
- [26] C. Brezinski and M. Redivo Zaglia. *Extrapolation methods. Theory and practice*. North-Holland, Amsterdam, 1991.
- [27] S. Cabay and L. W. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM J. Numer. Anal.*, 13:734–752, 1976.
- [28] C. Brezinski. Généralisation de la transformation de Shanks, de la table de Padé et de l’algorithme. *Calcolo*, 12:317–360, 1975.
- [29] R. P. Eddy. Extrapolation to the limit of a vector sequence. In P. C. C. Wang, editor, *Information Linkage Between Applied Mathematics and Industry*, pages 387–396. Academic Press, New York, 1979.
- [30] M. Mešina. Convergence acceleration for the iterative solution of $x = Ax + f$. *Comput. Methods Appl. Mech. Eng.*, 10:165–173, 1977.
- [31] M. D. Feit and Jr. J. A. Fleck. Solution of the Schrödinger equation by a spectral method II: Vibrational energy levels of triatomic molecules. *J. Chem. Phys.*, 78 (1):301–308, 1983.
- [32] M. D. Feit and Jr. J. A. Fleck. Wave packet dynamics and chaos in the Hénon-Heiles system. *J. Chem. Phys.*, 80 (6):2578–2584, 1984.
- [33] A. D. Bandrauk and H. Shen. Improved exponential split operator method for solving the time-dependent Schrödinger equation. *Chem. Phys. Lett.*, 176 (5):428–432, 1991.
- [34] A. D. Bandrauk and H. Shen. Higher order exponential split operator method for solving time-dependent Schrödinger equations. *Can. J. Chem.*, 70:555–559, 1992.
- [35] A. D. Bandrauk and H. Shen. Exponential split operator methods for solving coupled time-dependent Schrödinger equations. *J. Chem. Phys.*, 99 (2):1185–1193, 1993.
- [36] MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760-2098. *Matlab Version 5.2*.
- [37] G. H. Golub and Charles F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 1989.
- [38] The Numerical Algorithms Group Limited, Wilkinson House, Jordan Hill Road, Oxford. *NAG Fortran Library Manual, Mark 17*, 1995.
- [39] Waterloo Maple Inc., Canada. *Maple V Release 5.1*.
- [40] P. M. Morse. Diatomic molecules according to the wave mechanics. II. Vibrational levels. *Phys. Rev.*, 34:57–64, 1929.

- [41] M. Zhao, S. Jang, and S. A. Rice. Quantum wave packet dynamics of van der Waals fragmentation. *Internet Journal of Chemistry*, 35, 1998.
- [42] S. K. Gray, S. A. Rice, and D. W. Noid. The classical mechanics of vibrational predissociation: A model based study of phase space structure and its influence on fragmentation rates. *J. Chem. Phys.*, 84:3745–3752, 1986.
- [43] M. Zhao and S. A. Rice. Unimolecular fragmentation rate theory revisited: An improved classical theory. *J. Chem. Phys.*, 96:3542–3548, 1992.
- [44] M. Zhao and S. A. Rice. An approximate classical unimolecular reaction rate theory. *J. Chem. Phys.*, 96:6654–6665, 1992.
- [45] M. Zhao and S. A. Rice. Comment on the rate of vibrational predissociation of some RgCl_2 and RgICl molecules. *J. Chem. Phys.*, 96:7483–7489, 1992.
- [46] S. K. Gray. A periodically forced oscillator model of van der Waals fragmentation: Classical and quantum dynamics. *J. Chem. Phys.*, 87:2051–2061, 1987.
- [47] O. Sharafeddin and J. H. Zhang. A DVR based time-dependent wave packet treatment for reactive scattering. *Chem. Phys. Lett.*, 204 (1,2):190–196, 1993.
- [48] M. Zhao and S. A. Rice. Comment on the classical theory of the rate of isomerization. *J. Chem. Phys.*, 97:943–951, 1992.
- [49] M. Zhao and S. A. Rice. *J. Chem. Phys.*, 98:3824, 1993.
- [50] S. K. Gray and S. A. Rice. *Faraday Discuss. Chem. Sci.*, 82:307, 1986.
- [51] S. K. Gray. In N. Halberstadt and K. C. Janda, editors, *Dynamics of Polyatomic van der Waals Complexes*. Plenum Press, New York, 1990.
- [52] S. K. Gray and S. A. Rice. A scattering resonance description of very low energy collision induced vibrational relaxation. *J. Chem. Phys.*, 83:2818–2828, 1985.
- [53] S. K. Gray and S. A. Rice. Phase space bottlenecks and statistical theories of isomerization reactions. *J. Chem. Phys.*, 86:2020–2035, 1987.
- [54] S. A. Rice and M. Zhao. New approaches to a classical theory of unimolecular reaction rate. *Int. J. Quantum Chem.*, 58:593–635, 1996.
- [55] S. K. Gray and C. E. Wozny. Wave packet dynamics of van der Waals molecules: Fragmentation of NeCl_2 with three degrees of freedom. *J. Chem. Phys.*, 91:7671–7684, 1989.
- [56] S. K. Gray and C. E. Wozny. Fragmentation mechanisms from three-dimensional wave packet studies: Vibrational predissociation of NeCl_2 , HeCl_2 , NeICl , and HeICl . *J. Chem. Phys.*, 94:2817–2832, 1991.
- [57] S. Nordholm and S. A. Rice. A quantum ergodic theory approach to unimolecular fragmentation. *J. Chem. Phys.*, 62:157–168, 1975.
- [58] S. K. Gray, S. A. Rice, and M. J. Davis. Bottlenecks to unimolecular reactions and an alternative form for classical RRKM theory. *J. Phys. Chem.*, 90:3470–3482, 1986.
- [59] K. E. Johnson, L. Wharton, and D. H. Levy. The photodissociation lifetime of the van der Waals molecule I_2He . *J. Chem. Phys.*, 69:2719–2724, 1978.
- [60] J. E. Kenny, K. E. Johnson, W. Shafin, and D. H. Levy. The photodissociation of van der Waals molecules: Complexes of iodine, neon, and helium. *J. Chem. Phys.*, 72:1109–1119, 1980.
- [61] David W. Oxtoby and Stuart A. Rice. Nonlinear resonance and stochasticity in intramolecular energy exchange. *J. Chem. Phys.*, 65 (5):1676–1683, 1976.
- [62] Research Systems Inc., 4990 Pearl East Circle, Boulder, CO 80301. *IDL Version 5.0*.

- [63] E. J. Weniger. Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series. *Comput. Phys. Rep.*, 10:189–371, 1989.
- [64] Daniel Neuhauser and Michael Baer. The time-dependent Schrödinger equation: Application of absorbing boundary conditions. *J. Chem. Phys.*, 90 (8):4351–4355, 1989.
- [65] Á. Vibók and G. G. Balint-Kurti. Parametrization of complex absorbing potentials for time-dependent quantum dynamics. *J. Phys. Chem.*, 96:8712–8719, 1992.

Erklärung

Hiermit erkläre ich, daß ich die Diplomarbeit selbständig angefertigt und keine Hilfsmittel außer den in der Arbeit angegebenen benutzt habe.

Regensburg, den 16. November 1999

.....
(Unterschrift)